ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Bài 5:
Một số ứng dụng học sâu trong xử lý ngôn ngữ tự nhiên
(Phần 2)

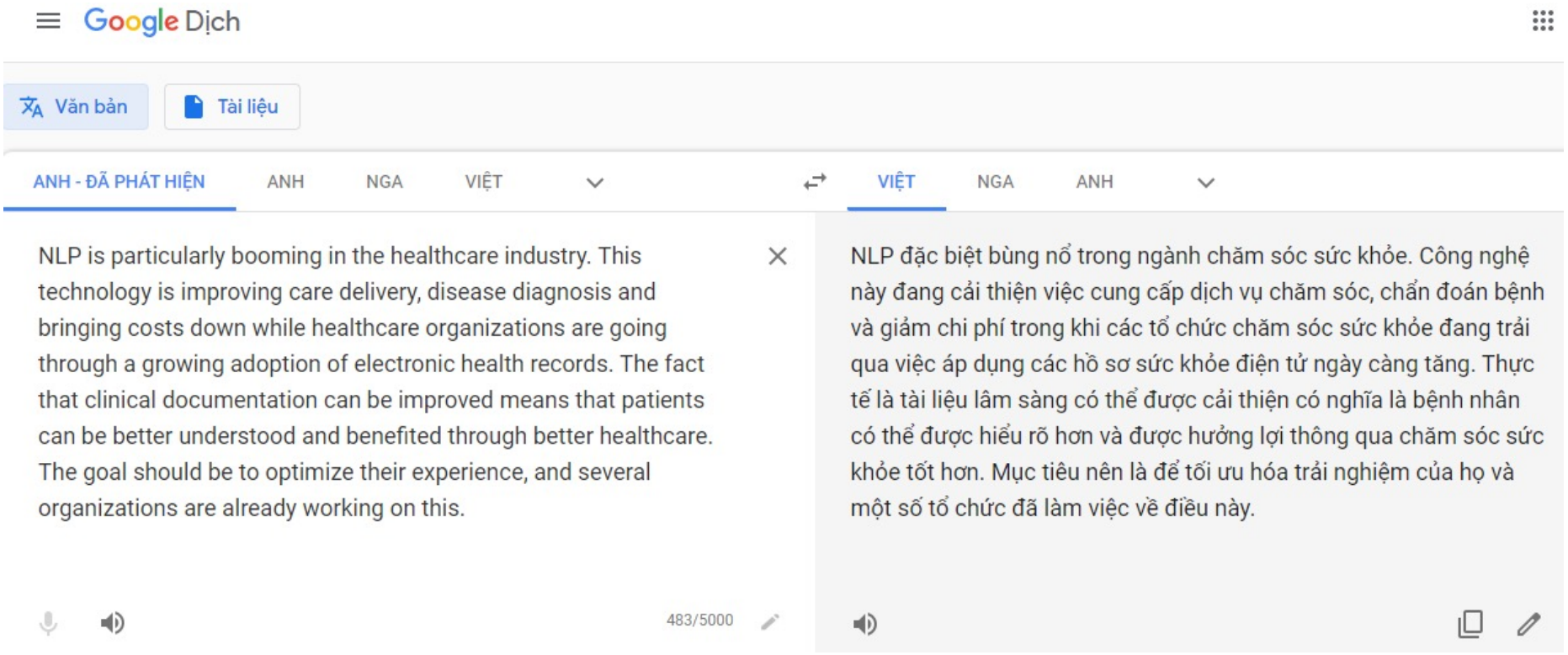
Nội dung

1. Giới thiệu về bài toán dịch máy
2. Mô hình NMT
3. Cơ chế chú ý (attention)

Giới thiệu về bài toán dịch máy

Dịch máy

- Google translate



Dịch máy – Machine Translation

- **Dịch máy (MT)** là thao tác dịch một câu x từ một ngôn ngữ (gọi là ngôn ngữ nguồn) sang một câu y trong ngôn ngữ khác (gọi là ngôn ngữ đích)

x: *L'homme est né libre, et partout il est dans les fers*

y: *Man is born free, but everywhere he is in chains*

Dịch máy – Machine Translation

- Bắt đầu từ những năm 1950
- Dịch từ Nga sang Anh (nhu cầu xuất phát từ chiến tranh lạnh)
- Hệ thống dịch chủ yếu theo quy tắc (rule-based), dùng từ điển để ánh xạ các từ tiếng Nga sang tiếng Anh

1 minute video showing 1954 MT:
https://youtu.be/K-HfpsHPmvw

Dịch máy dựa trên luật

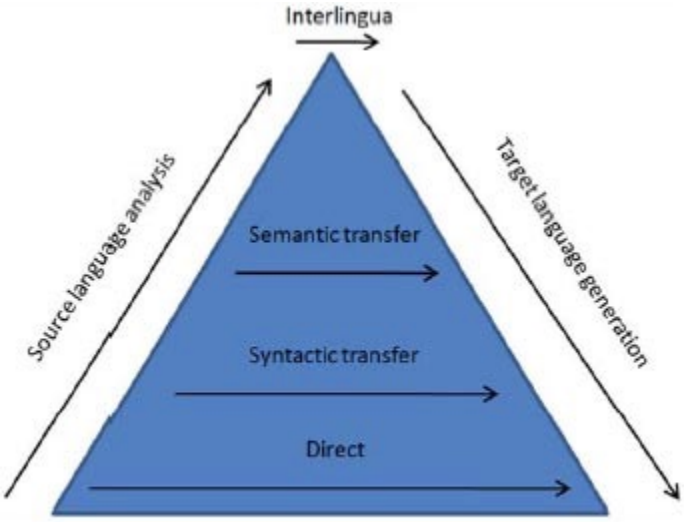
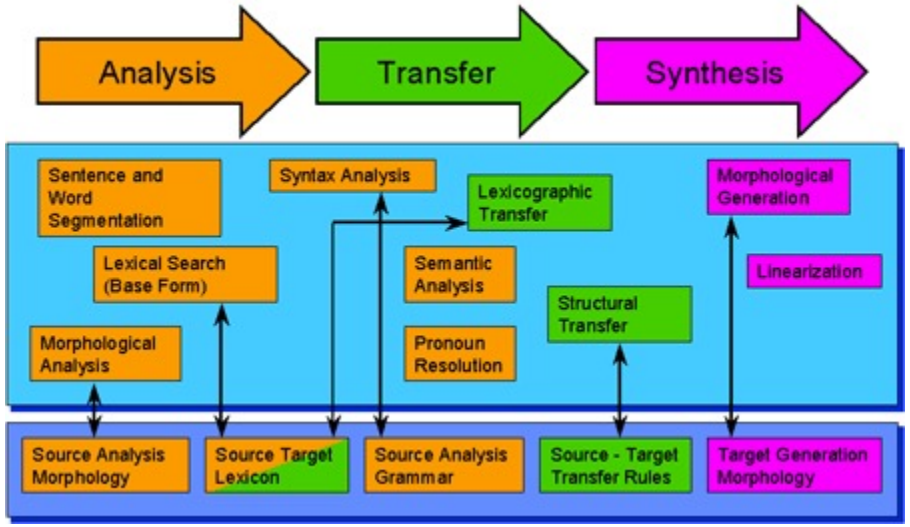


Figure 1: The Vauquois triangle



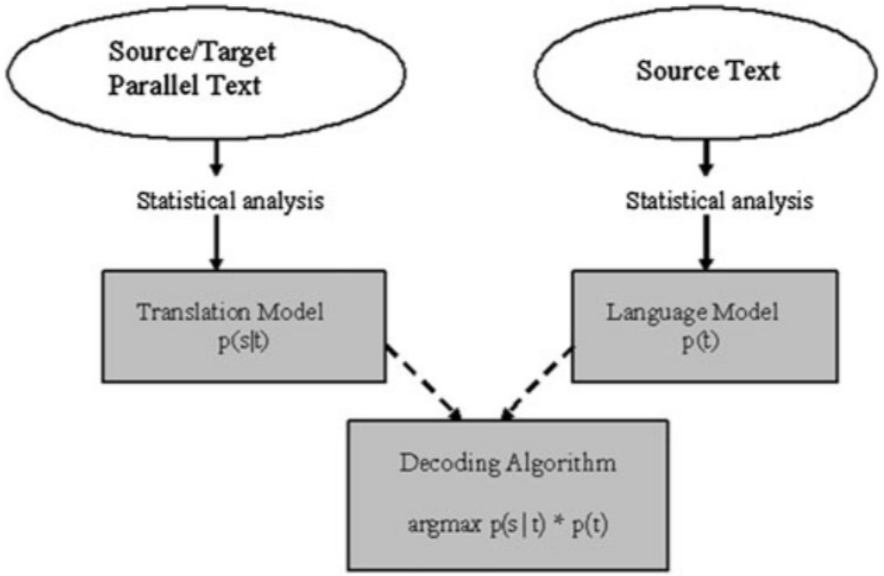
© https://www.coroflot.com/tuyenduong/machine-translation

Dịch máy dựa trên luật

- Nhiều xử lý thủ công và sức người
  - Từ điển ánh xạ từ Nguồn – Đích
  - Các luật chuyển đổi (lexical, structure)
  - Các luật hình thái học (Morphological rules)
- Chất lượng thấp

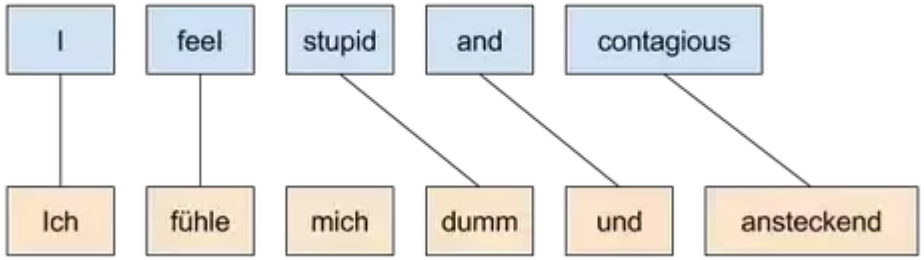
Dịch máy thống kê (1990s-2010s)

- Dịch máy thống kê (statistical machine translation) học một mô hình xác suất từ dữ liệu
- Mục tiêu: Tìm kiếm câu tốt nhất ở ngôn ngữ đích, từ câu đầu vào ở ngôn ngữ nguồn



Một cách mô hình hoá P(s|t)

- Giả định: Gióng mỗi từ trong câu nguồn với các từ trong câu đích
- Vector gióng (alignment vector) a = [1,2,4,5,6]
- Mục tiêu: Tìm một cách gióng sao cho cực đại hoá P(s,a|t)

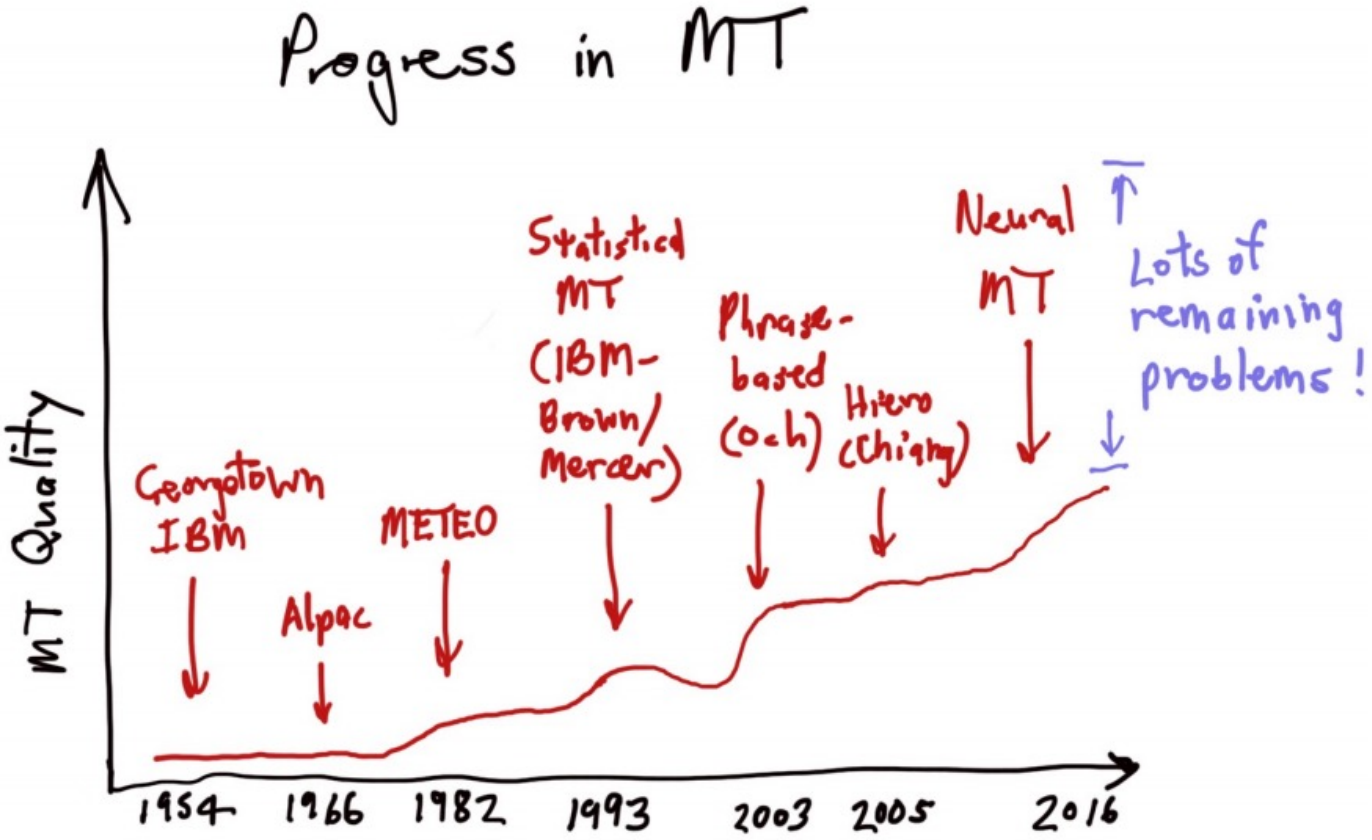


© Vasily Konovalov, MSc Natural Language Processing

Nhược điểm của dịch máy thống kê

- Các hệ thống tốt nhất theo hướng tiếp cận này rất phức tạp, mỗi hệ thống chứa nhiều mô-đun nhỏ được thiết kế độc lập nhau
  - Vẫn không đạt được hiệu năng như con người
- Cần nhiều xử lý thủ công và sức người
  - Kỹ nghệ đặc trưng (feature engineering)
  - Tài nguyên bên ngoài (extra resources)
- Chi phí bảo trì cao, khi chuyển sang cặp ngôn ngữ khác phải làm lại thủ công từ đầu, không tái sử dụng được sức người

Tiến bộ trong dịch máy

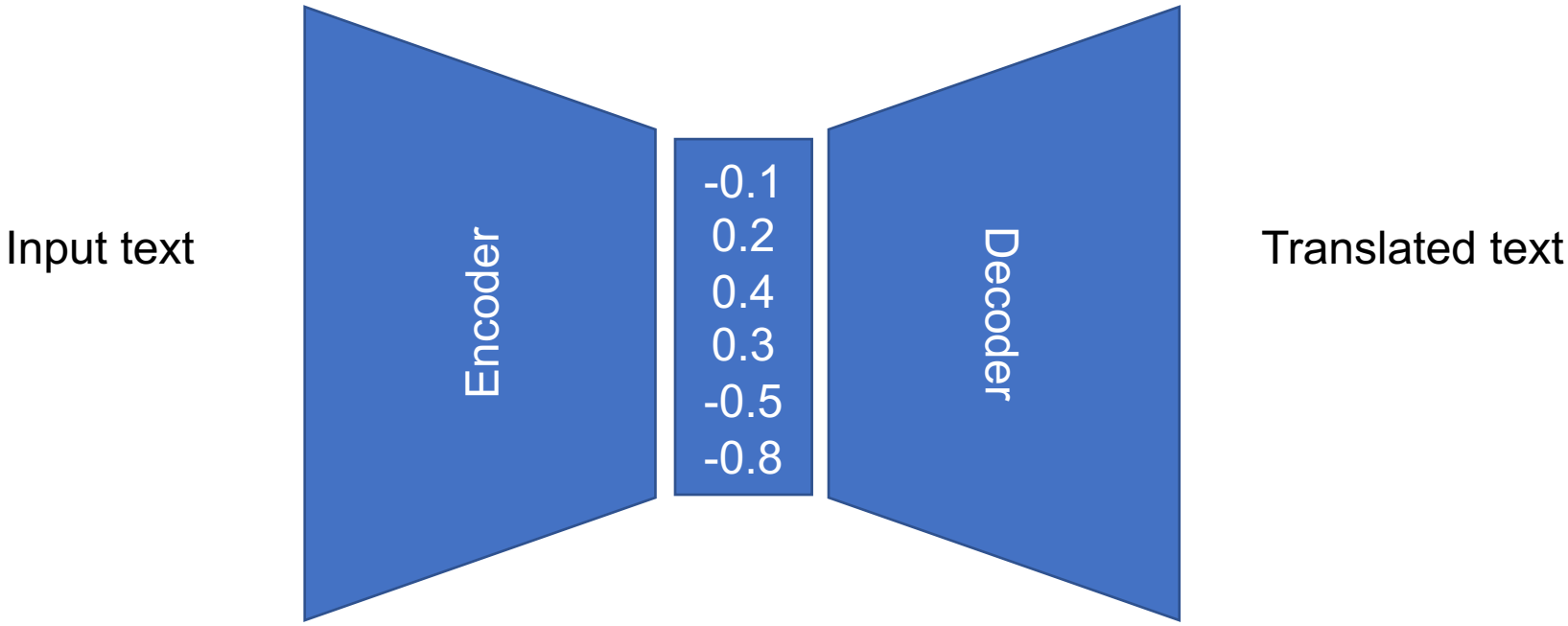


© https://sites.google.com/site/acl16nmt/home

Mô hình NMT
(Neural Machine Translation)

Neural Machine Translation is the approach of modeling the entire MT process via one big artificial neural network (ACL 2016)

Mô hình sequence-to-sequence

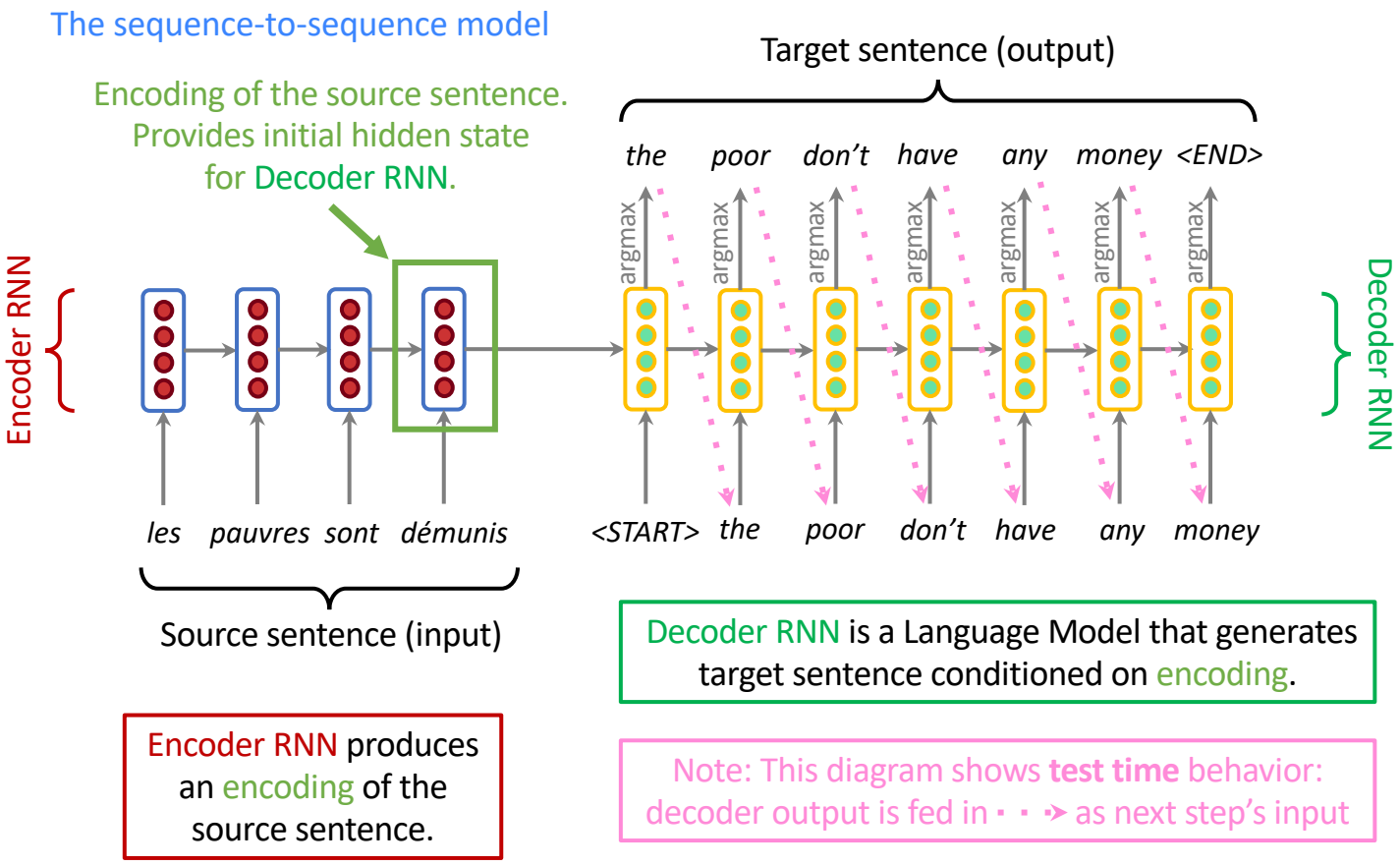


- Encoder RNN sinh ra "thông tin mã hóa" (encoding) của câu nguồn
- Decoder RNN sinh ra câu đích dựa trên thông tin mã hóa của câu nguồn

Mô hình sequence-to-sequence

- Mô hình seq2seq có thể sử dụng cho nhiều bài toán khác như:
  - Tóm lược văn bản (văn bản dài → văn bản ngắn)
  - Hội thoại (câu nói trước → câu nói tiếp theo)
  - Sinh code (ngôn ngữ tự nhiên → code python)
  - …

Neural machine translation (NMT)



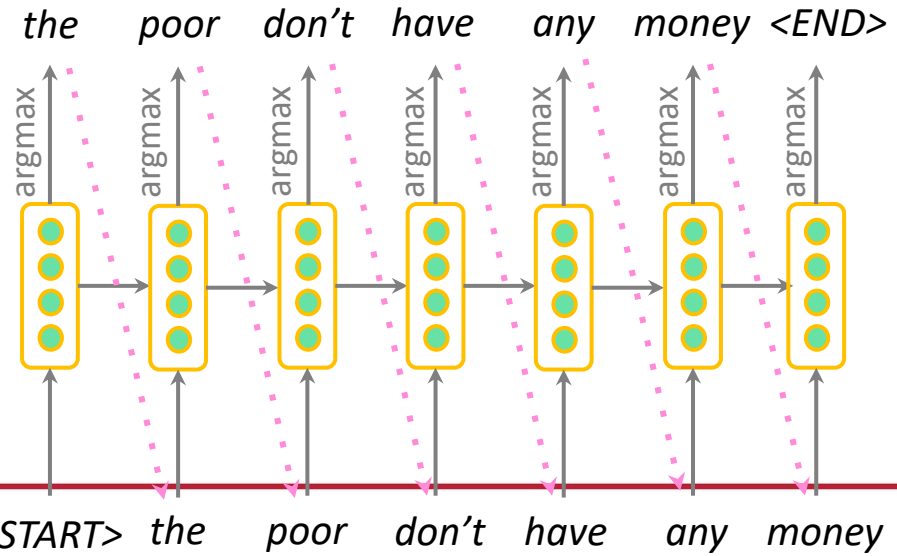
© Tensorflow for deep learning research. Standford

Xem dịch máy như mô hình ngôn ngữ có điều kiện

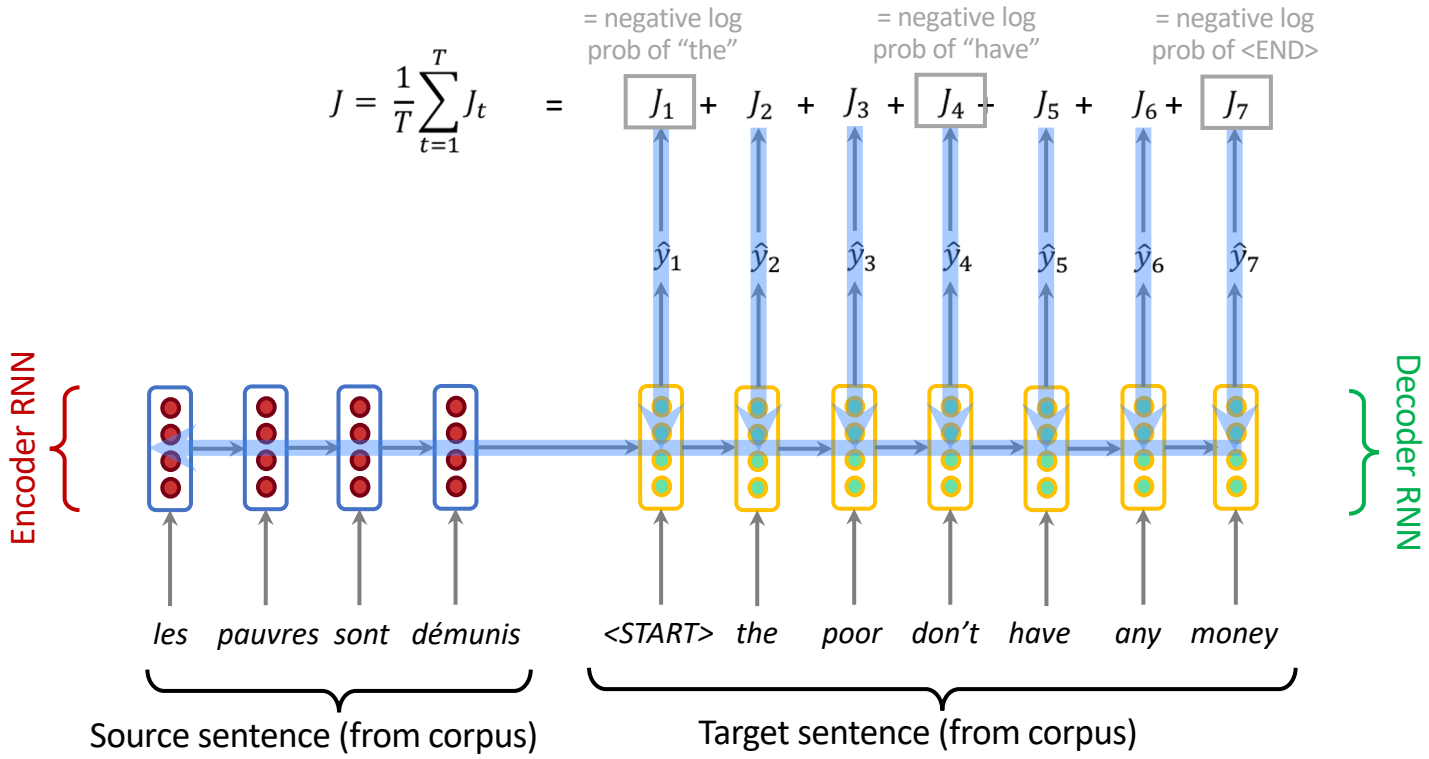
- NMT tính trực tiếp $P(y|x)$

$$P(y|x) = P(y_1|x)\,P(y_2|y_1,x)\,P(y_3|y_1,y_2,x)\ldots P(y_T|y_1,\ldots,y_{T-1},x)$$

- Mô hình ngôn ngữ có điều kiện (conditional language model)
  - Mô hình ngôn ngữ: dự đoán một từ dựa trên ngữ cảnh của các từ xung quanh
  - Có điều kiện: sự dự đoán được dựa trên thêm điều kiện từ câu nguồn



Huấn luyện mô hình seq2seq

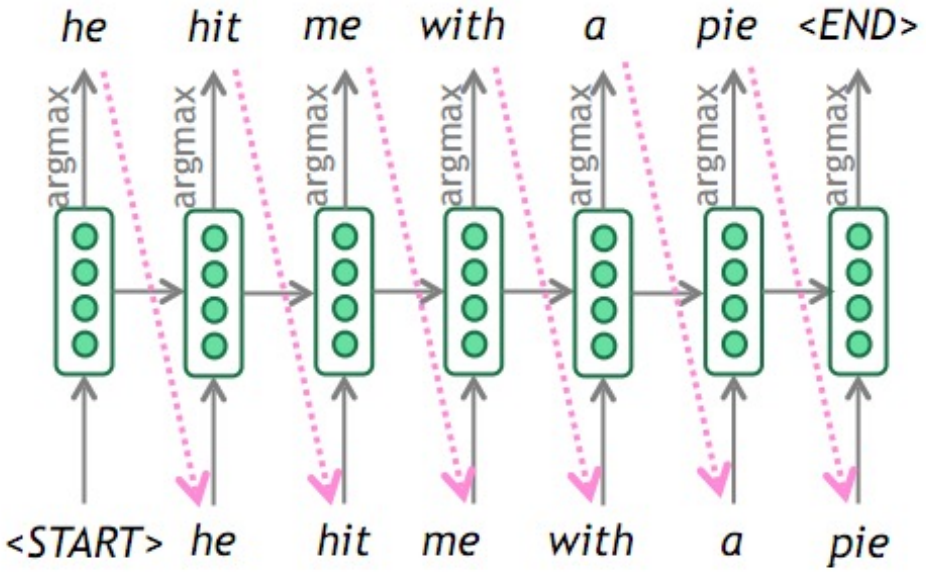


Seq2seq is optimized as a **single system.**
Backpropagation operates *"end to end"*.

© Tensorflow for deep learning research. Standford

Mô hình sequence-to-sequence

- Giải mã ra câu đích bằng cách lấy argmax tại từng bước
- Đây là cách giải mã tham lam
- Nếu lỡ sai ở một bước nào đó là sẽ sai luôn các bước sau, không có cách nào quay lại để sửa.



Điểm BLEU (Bilingual evaluation understudy)

- BLEU tính độ tương đồng giữa câu dịch sinh ra bởi mô hình và câu nhãn, do người dịch
  - Đo độ chính sác của các N-gram (N từ 1 tới 4)
  - Phạt các câu dịch quá ngắn

$$\text{BLEU} = \min\left(1, \frac{\textit{output-length}}{\textit{reference-length}}\right) \left(\prod_{i=1}^{4} \textit{precision}_i\right)^{\frac{1}{4}}$$

BLEU Score	Interpretation
30 - 40	Understandable to **good** translations
40 - 50	**High** quality translations
50 - 60	Very **high** quality, adequate, and fluent translations
> 60	Quality often **better** than human

Phương pháp tìm kiếm

Phương pháp tìm kiếm

- Ta mong muốn tìm được câu đích y (độ dài T) cực đại hóa xác suất hậu nghiệm:

$$P(y|x) = P(y_1|x)\,P(y_2|y_1,x)\,P(y_3|y_1,y_2,x)\ldots,P(y_T|y_1,\ldots,y_{T-1},x)$$
$$= \prod_{t=1}^{T} P(y_t|y_1,\ldots,y_{t-1},x)$$

- Ta có thể tính với tất cả các phương án của y.
- Độ phức tạp V^T với V là kích thước tập từ vựng.

Tìm kiếm chùm – beam search

- Ý tưởng: Tại mỗi bước giải mã, ta duy trì *k* phương án bộ phận có xác suất xảy ra cao nhất (gọi là các giả thuyết)
- *k* là kích thước chùm (beam size)
- Một giả thuyết $y_1, y_2, \ldots, y_t$ có điểm bằng log giá trị xác suất của nó:

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

- Tất cả điểm score đều âm, điểm càng cao càng tốt
- Ta sẽ giữ lại *k* giả thuyết có điểm score cao nhất tại mỗi bước
- Tìm kiếm chùm không đảm bảo tìm được lời giải tối ưu
- Nhưng hiệu quả hơn rất nhiều so với phương pháp duyệt

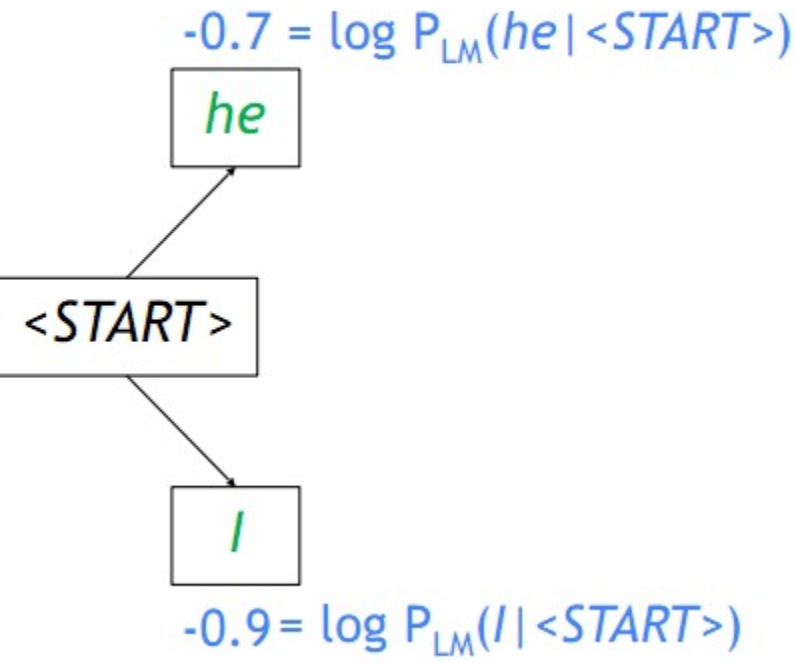
Ví dụ beam search với k = 2

- Tính toán phân phối xác suất từ tiếp theo

<START>

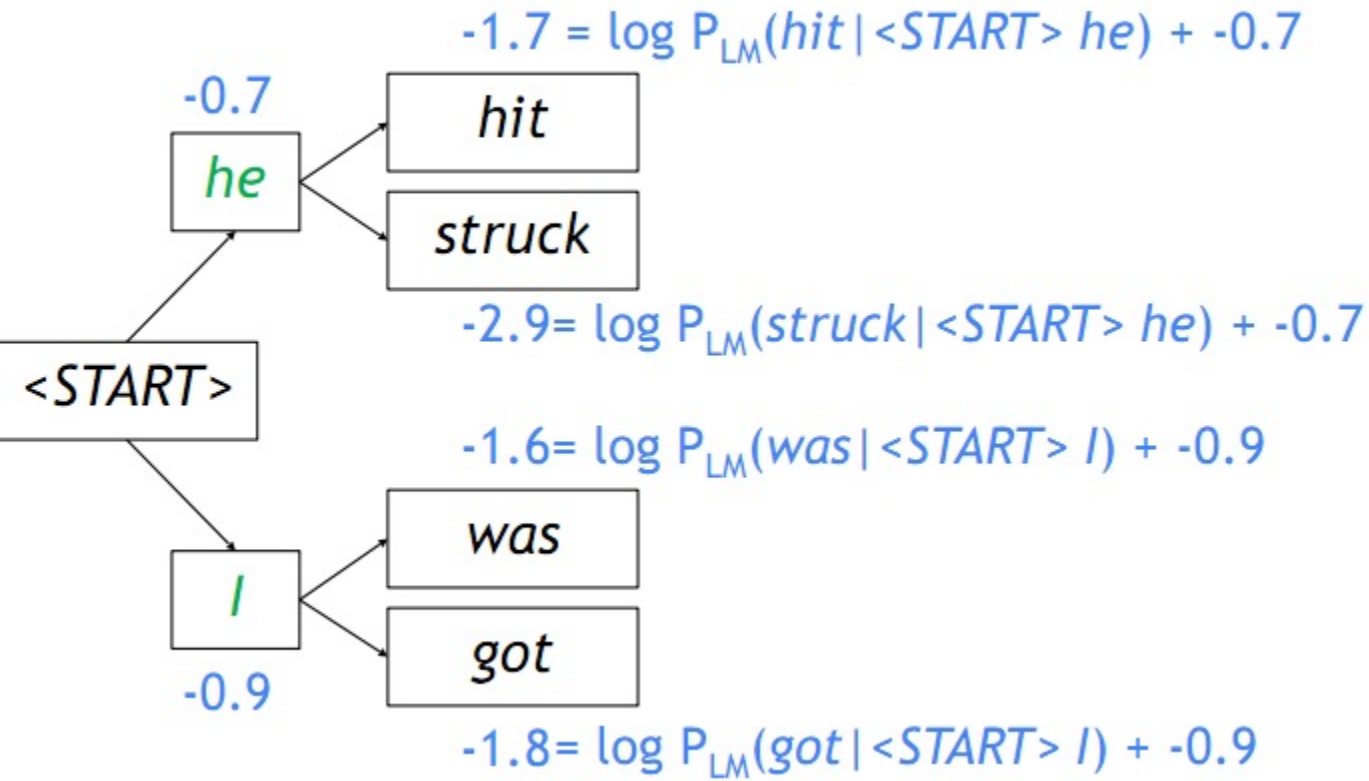
Ví dụ beam search với k = 2

- Giữ hai phương án với điểm cao nhất



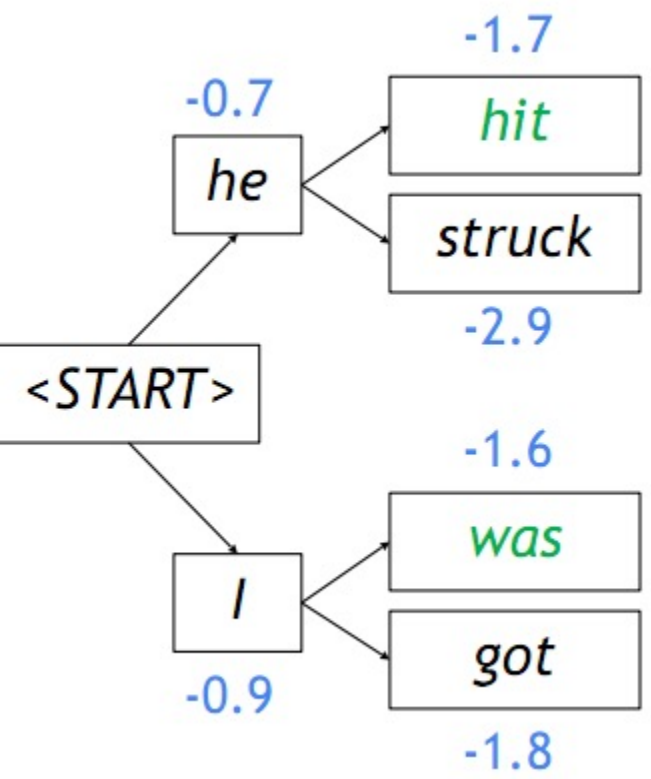
Ví dụ beam search với k = 2

- Với mỗi giả thuyết tìm tiếp *k* giả thuyết tiếp theo có điểm cao nhất



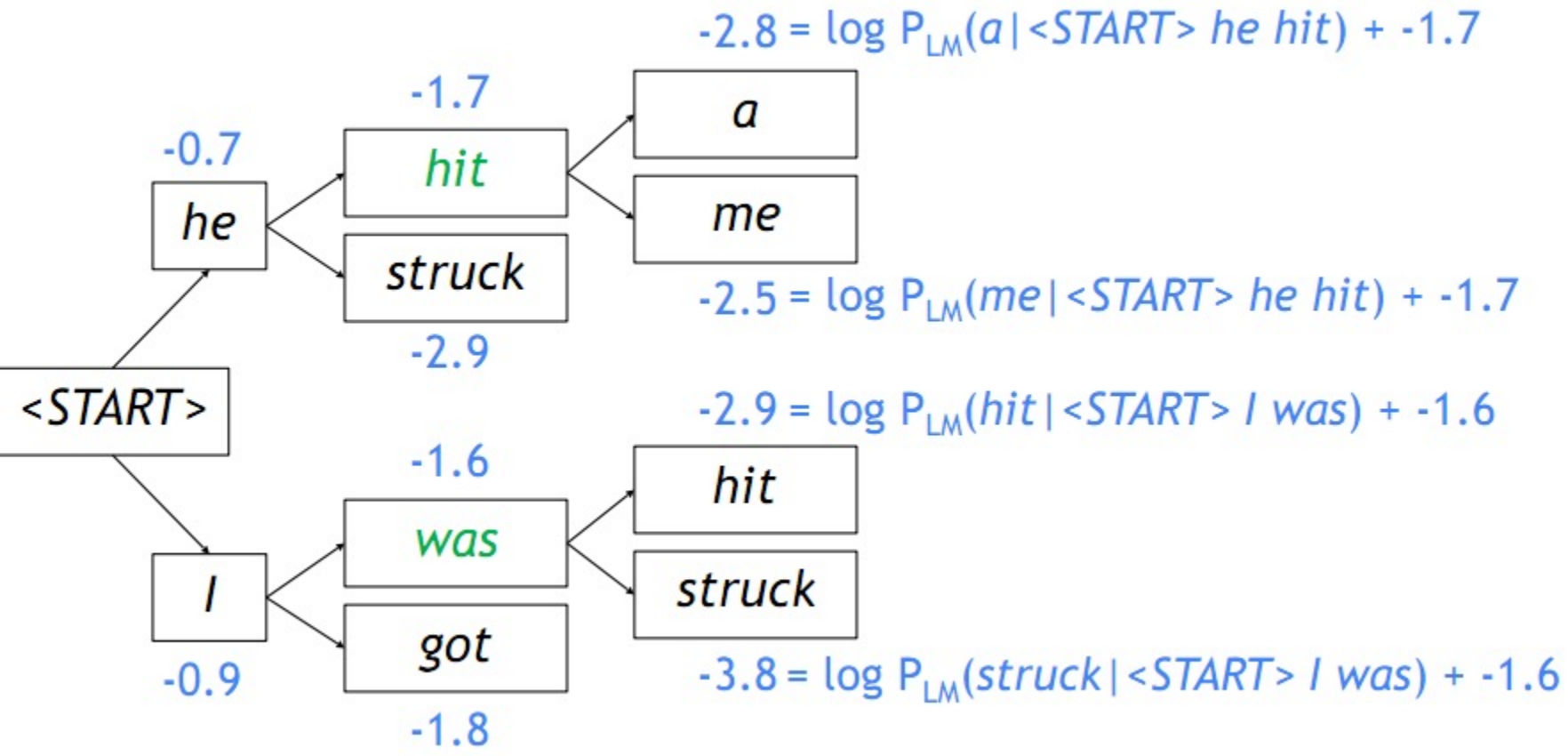
Ví dụ beam search với k = 2

- Trong k^2 giả thuyết mới ta chỉ giữ lại k giả thuyết điểm cao nhất



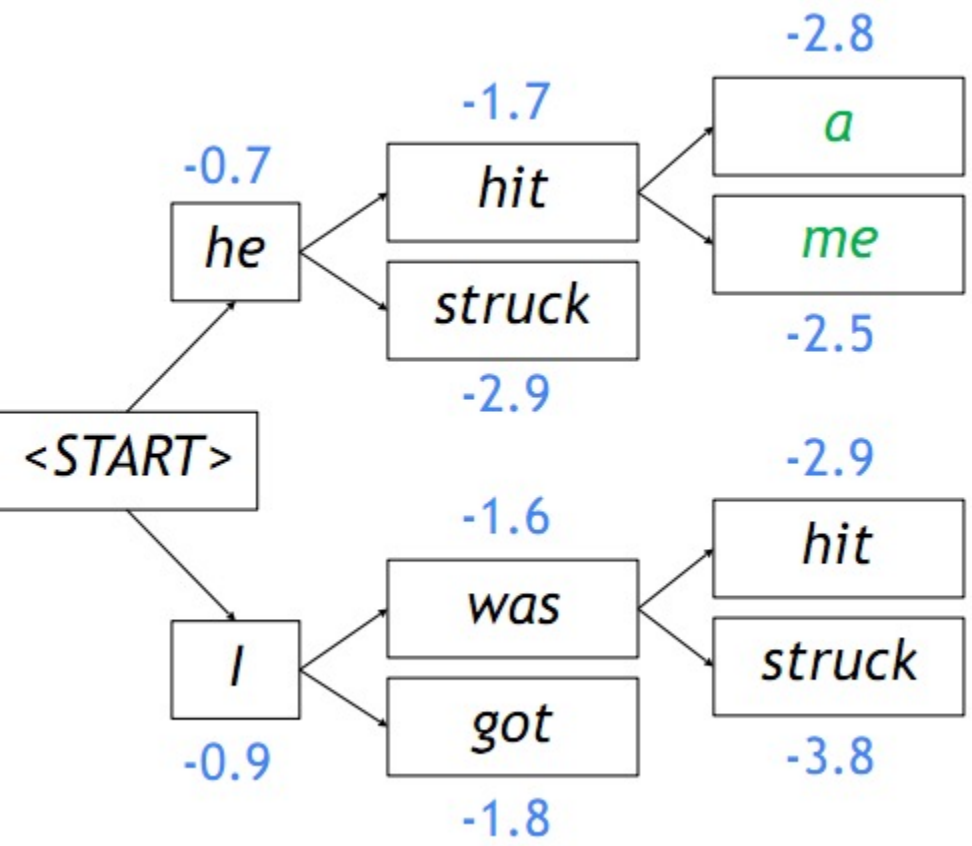
Ví dụ beam search với k = 2

- Với mỗi giả thuyết tìm tiếp *k* giả thuyết tiếp theo có điểm cao nhất



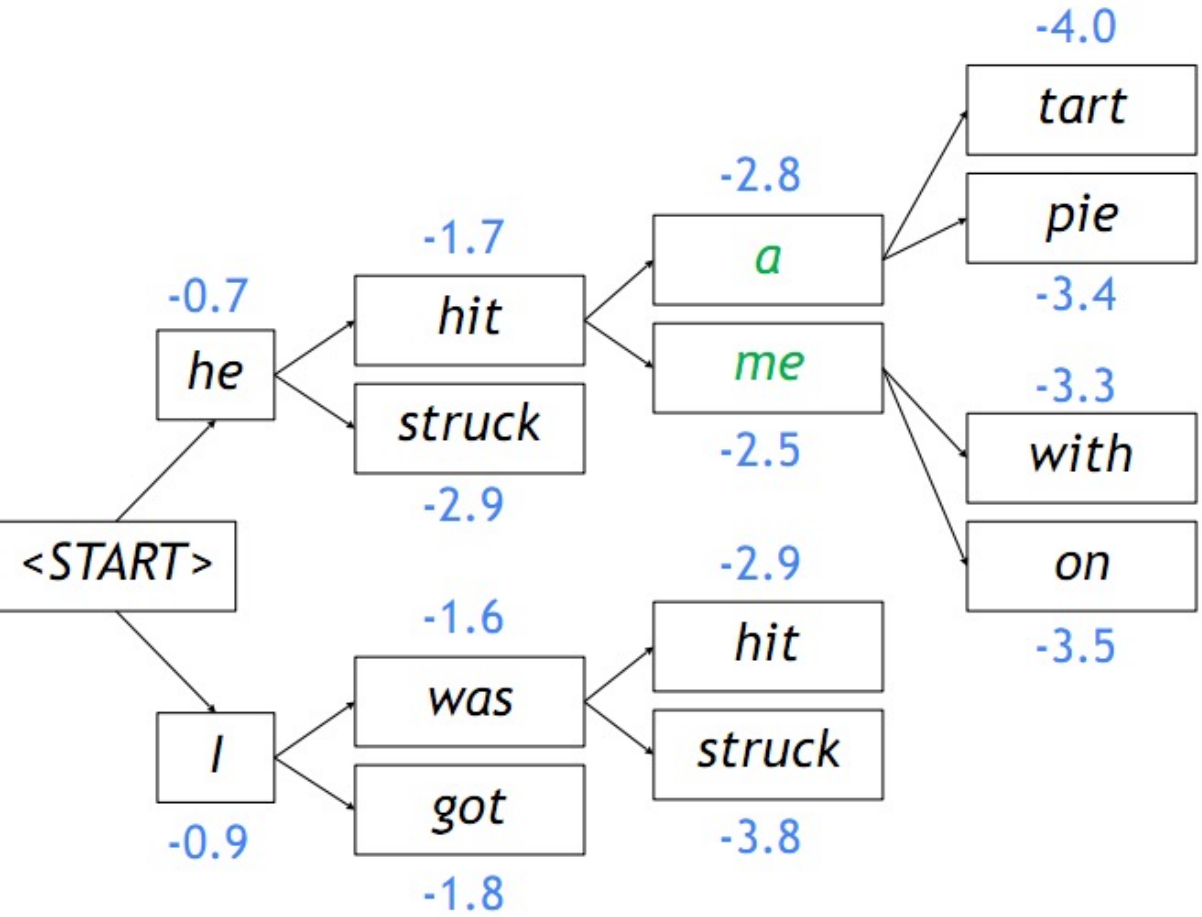
Ví dụ beam search với k = 2

- Trong k^2 giả thuyết mới ta chỉ giữ lại k giả thuyết điểm cao nhất



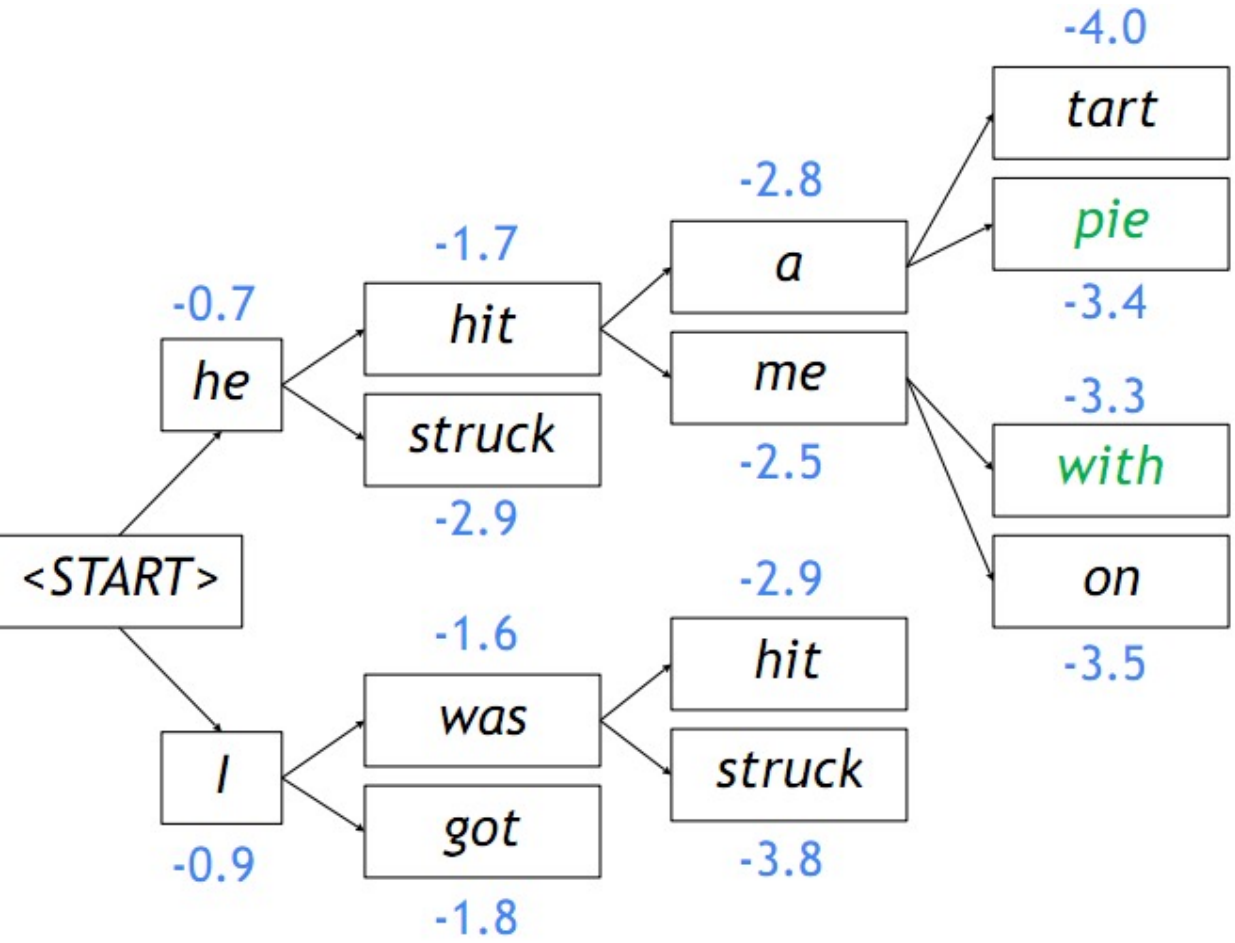
Ví dụ beam search với k = 2

- Với mỗi giả thuyết tìm tiếp *k* giả thuyết tiếp theo có điểm cao nhất



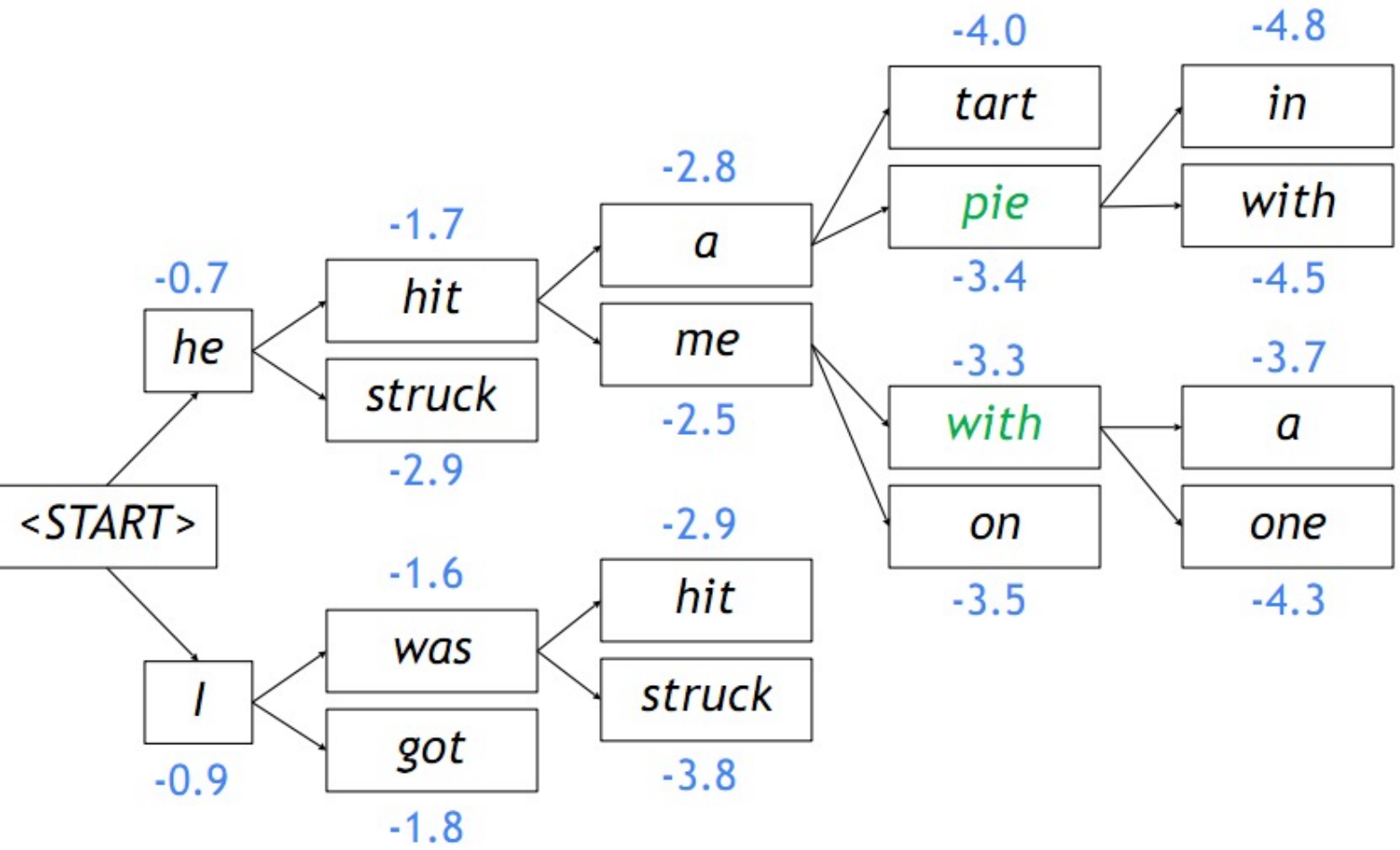
Ví dụ beam search với k = 2

- Trong k^2 giả thuyết mới ta chỉ giữ lại k giả thuyết điểm cao nhất



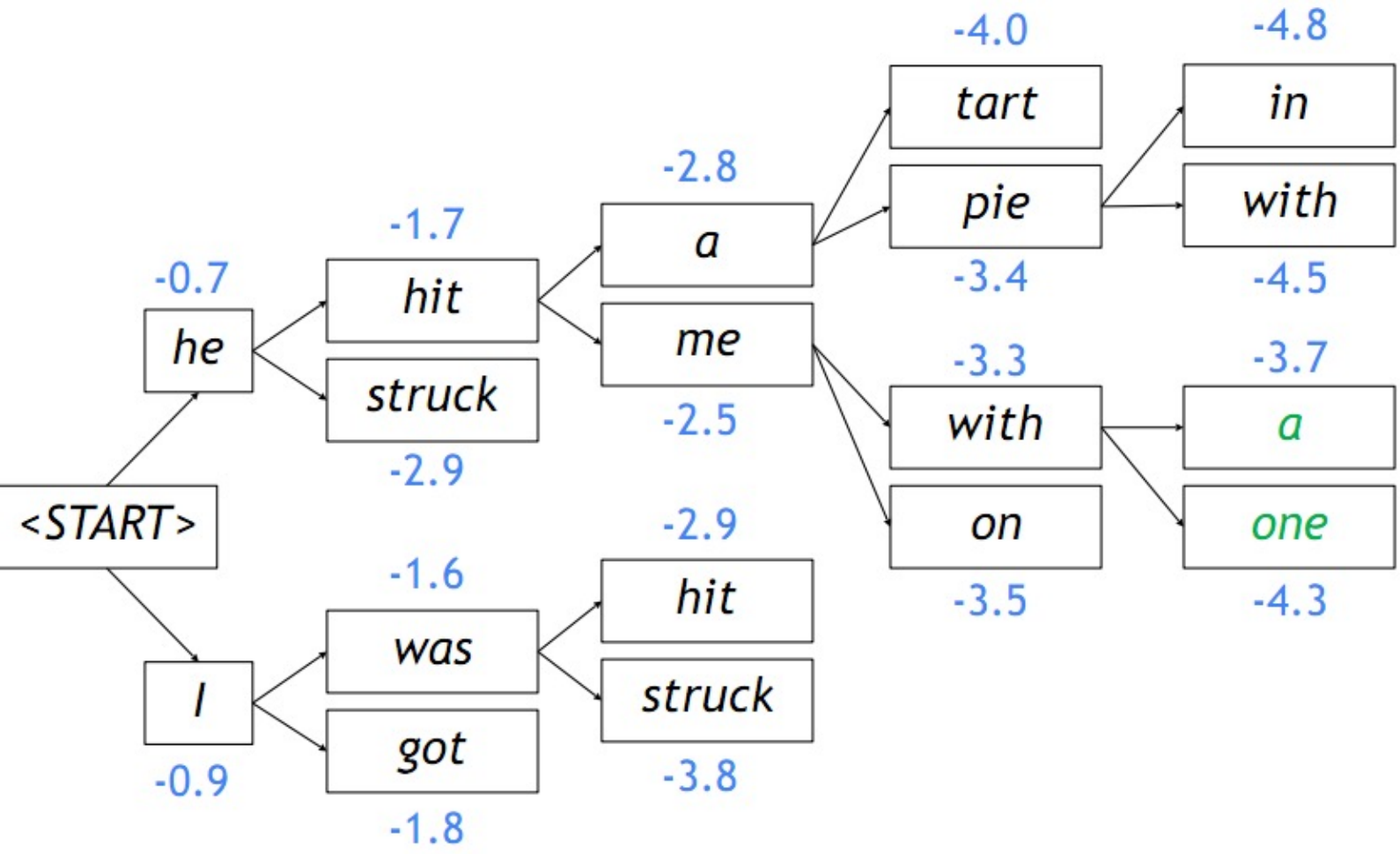
Ví dụ beam search với k = 2

- Với mỗi giả thuyết tìm tiếp *k* giả thuyết tiếp theo có điểm cao nhất



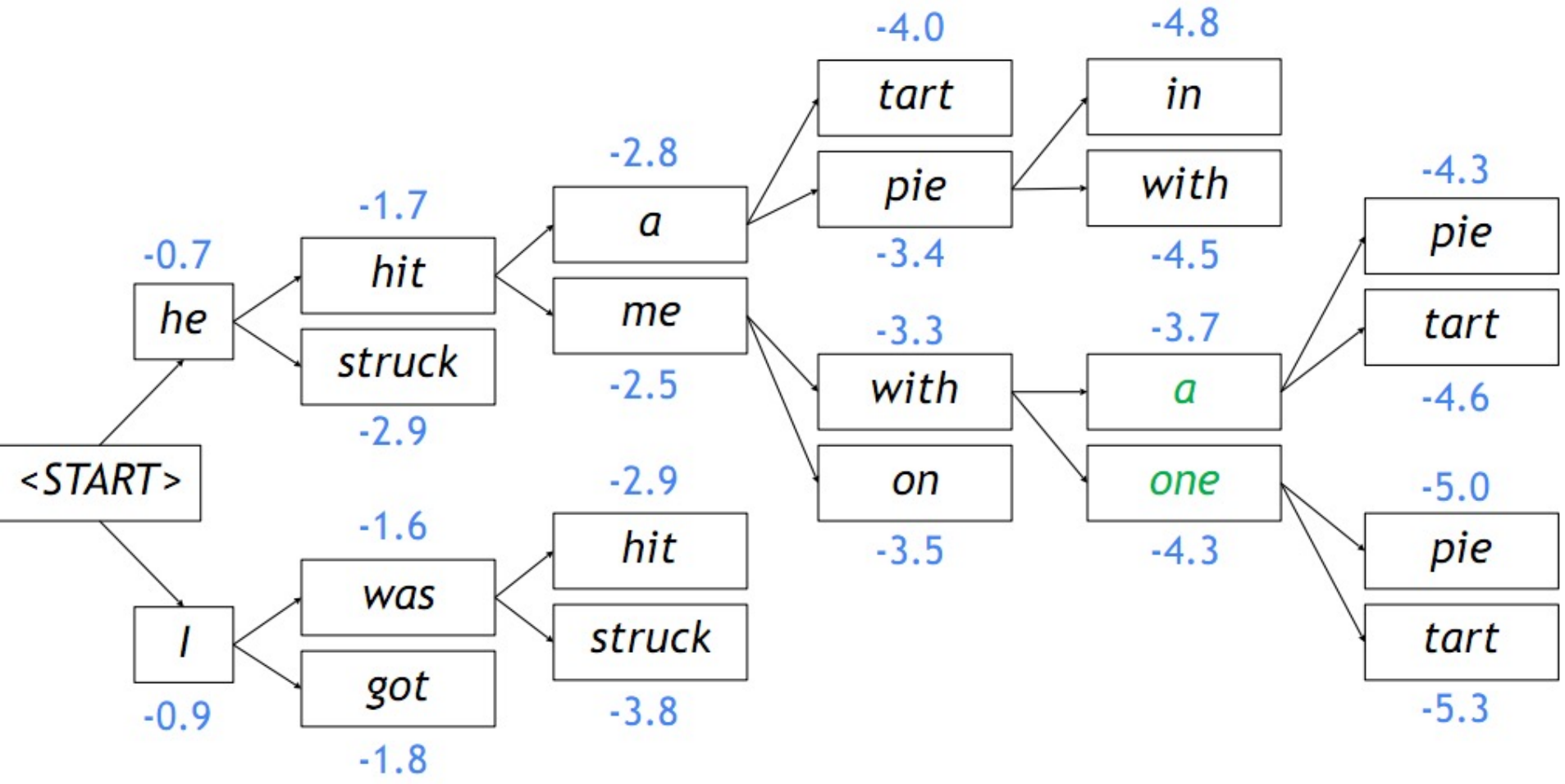
Ví dụ beam search với k = 2

- Trong k^2 giả thuyết mới ta chỉ giữ lại k giả thuyết điểm cao nhất



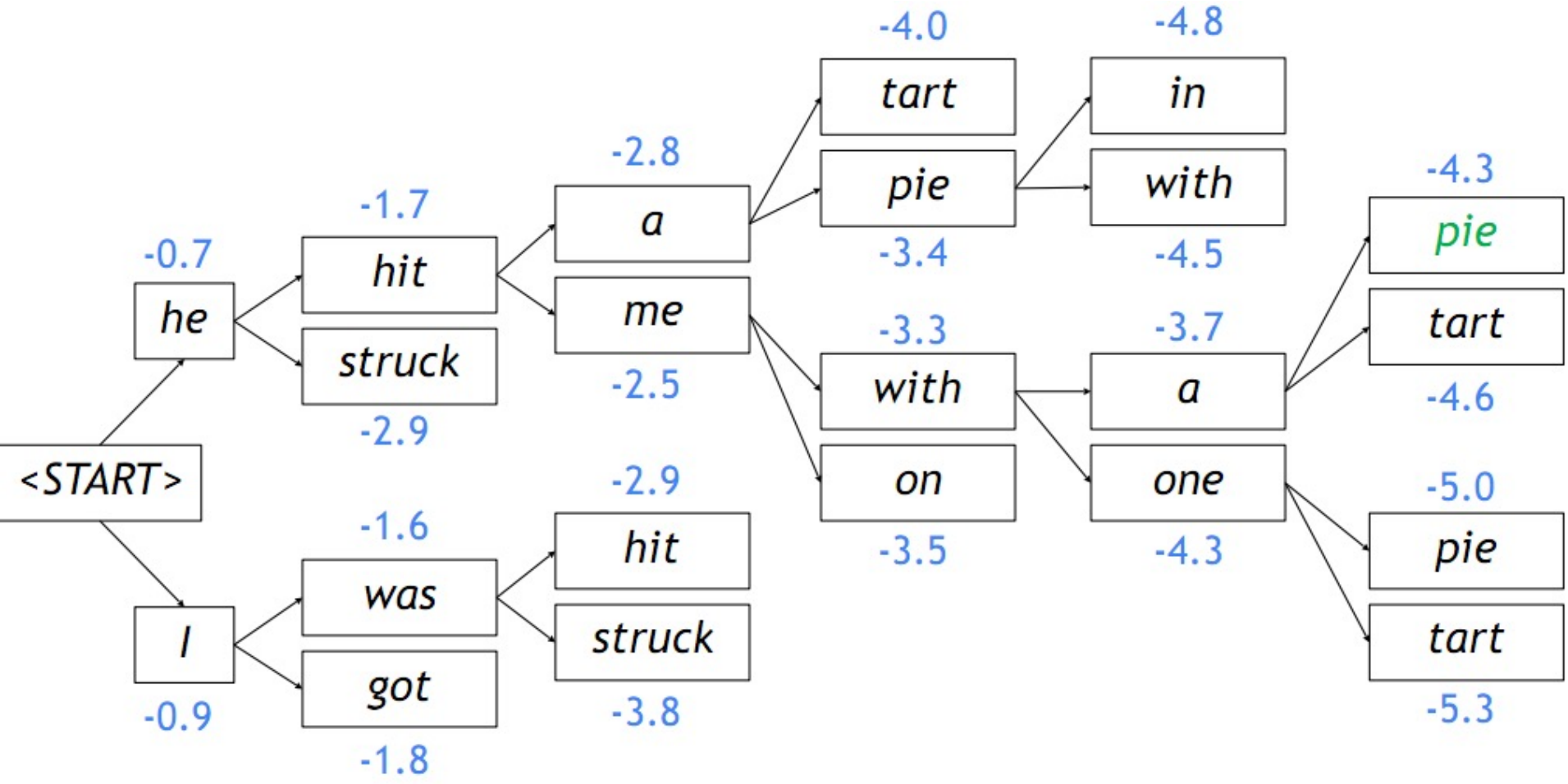
Ví dụ beam search với k = 2

- Với mỗi giả thuyết tìm tiếp *k* giả thuyết tiếp theo có điểm cao nhất



Ví dụ beam search với k = 2

- Giả thuyết có điểm cao nhất là lời giải cần tìm!



Điều kiện dừng beam search

- Trong giải mã tham lam, thường dừng khi mô hình sinh ra token <END>
- Ví dụ: <START> he hit me with a pie <END>
- Đối với beam search, các giả thuyết khác nhau có thể sinh ra token <END> tại các thời điểm khác nhau
- Khi một giả thuyết sinh ra <END> ta gọi giả thuyết đó được hoàn thành và đặt nó sang một bên để tiếp tục tìm các giả thuyết khác
- Thường sẽ dừng beam search khi:
  - Hoặc là đạt đến bước *T* cho trước
  - Hoặc khi đã tìm ra ít nhất *n* giả thuyết hoàn thành

Kết thúc beam search

- Khi tìm xong một tập các giả thuyết hoàn thành thì chọn giả thuyết nào?
- Vấn đề: giả thuyết càng dài điểm càng thấp

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

- Phương án giải quyết: Chuẩn hóa điểm theo chiều dài giả thuyết

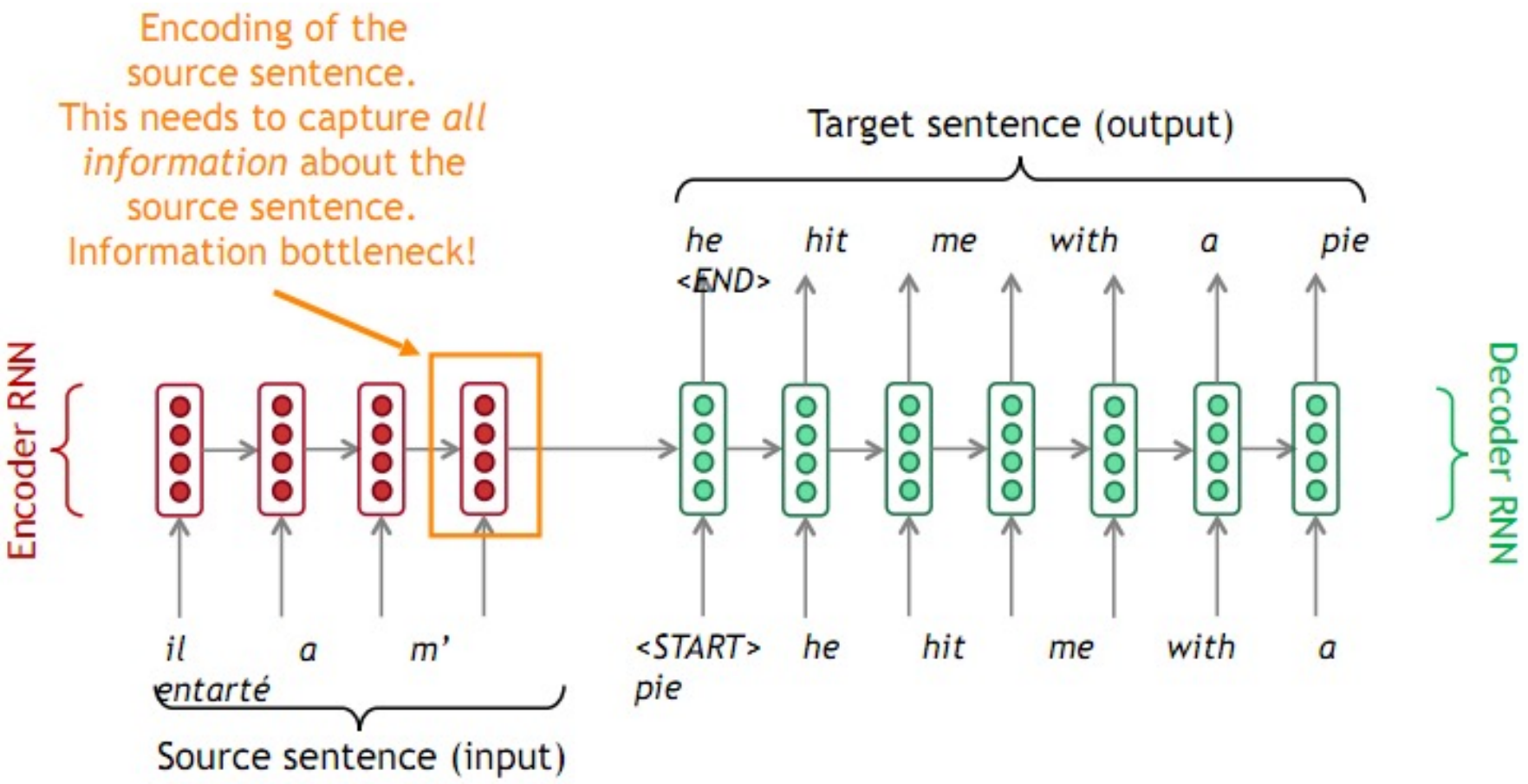
$$\frac{1}{t} \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

So sánh NMT và SMT

- **Ưu điểm** NMT so với SMT:
  - Hiệu năng tốt hơn: dịch trôi chảy hơn, dùng ngữ cảnh tốt hơn…
  - Chỉ dùng một mạng duy nhất nên có thể huấn luyện end-to-end, không cần tối ưu các mô-đun độc lập nào khác
  - Cần ít sức người hơn: không cần trích xuất đặc trưng thủ công, cùng một phương pháp có thể tái sử dụng cho nhiều cặp ngôn ngữ khác nhau
- **Nhược điểm** NMT so với SMT:
  - NMT khó giải thích hơn, khó gỡ rối
  - NMT khó kiểm soát. Ví dụ: muốn đưa một quy tắc hay gợi ý dịch cho NMT là không dễ dàng.

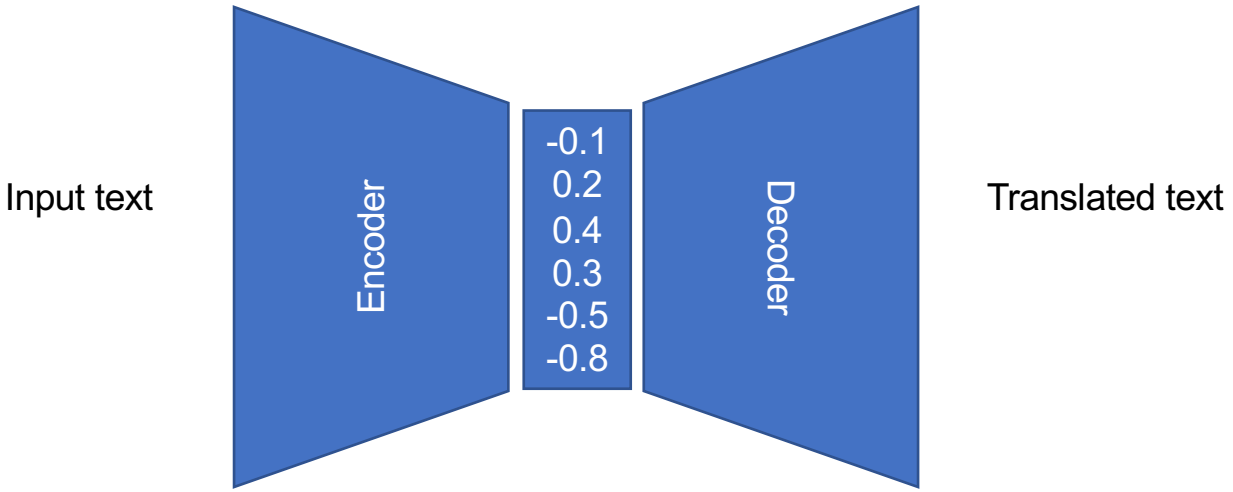
Cơ chế chú ý
(Attention mechanism)

Nút thắt cổ chai của mô hình seq2seq

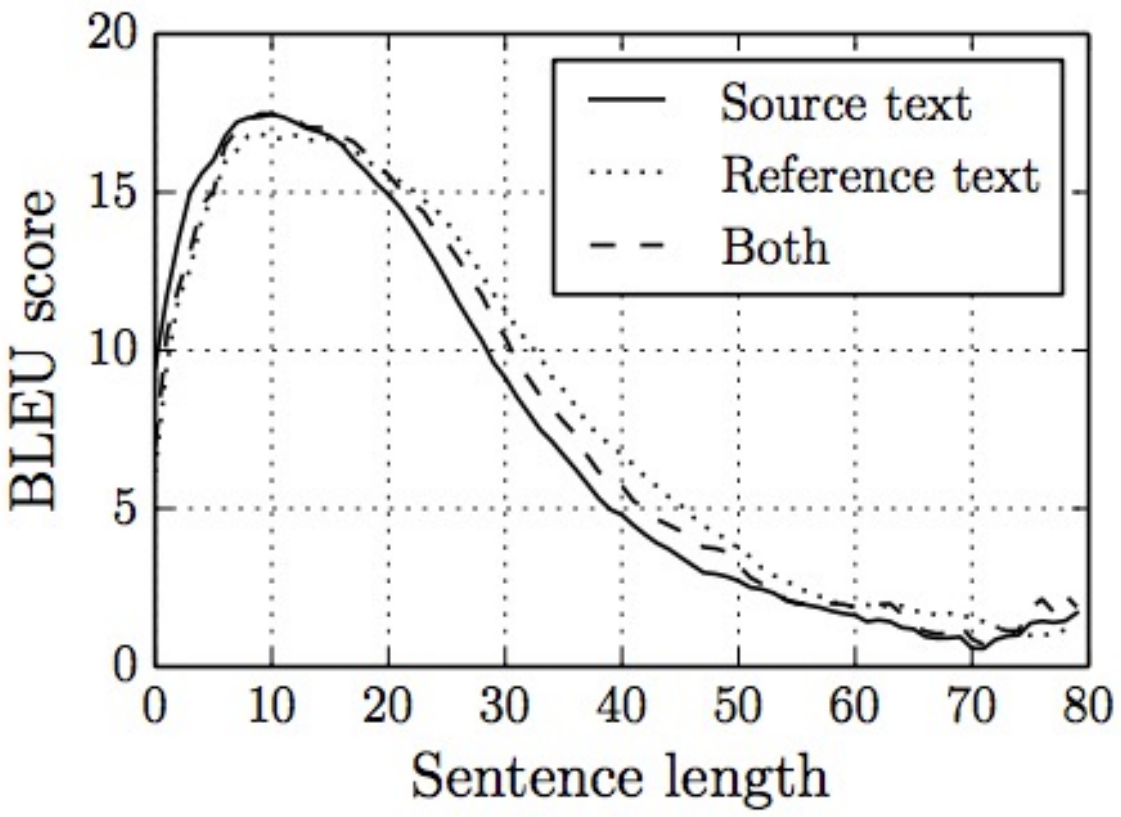


Ví dụ dịch câu dài

- Machine learning has turned out to be a very useful tool for translation, but it has a few weak spots. The tendency of translation models to do their work word by word is one of those, and can lead to serious errors.
- L'apprentissage automatique s'est révélé être un outil très utile pour la traduction, mais il comporte quelques points faibles. La tendance des modèles de traduction à faire leur travail mot à mot en fait partie et peut entraîner de graves erreurs.



Hiệu năng của mô hình vs. độ dài câu

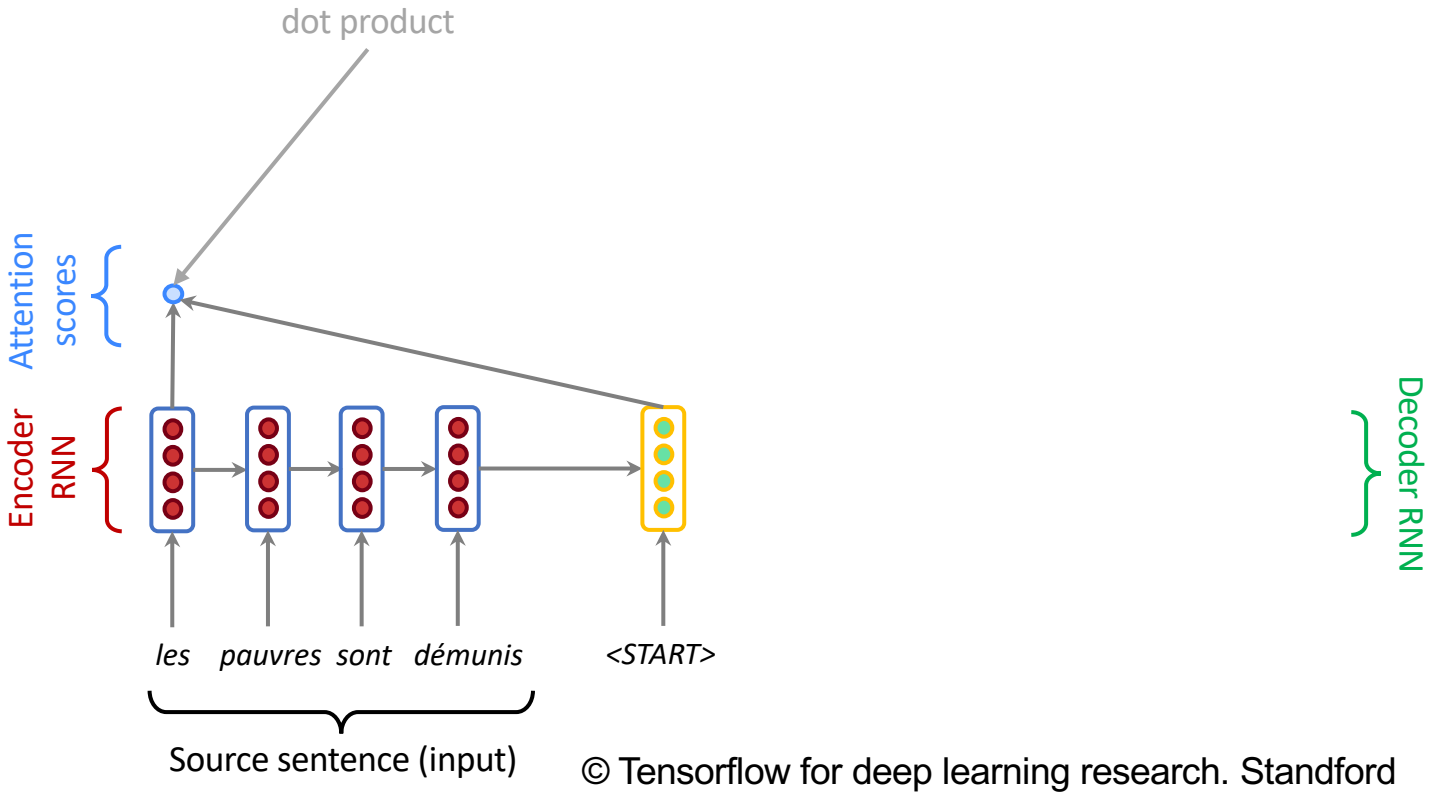


© On the Properties of Neural Machine Translation: Encoder-Decoder Approaches

Attention

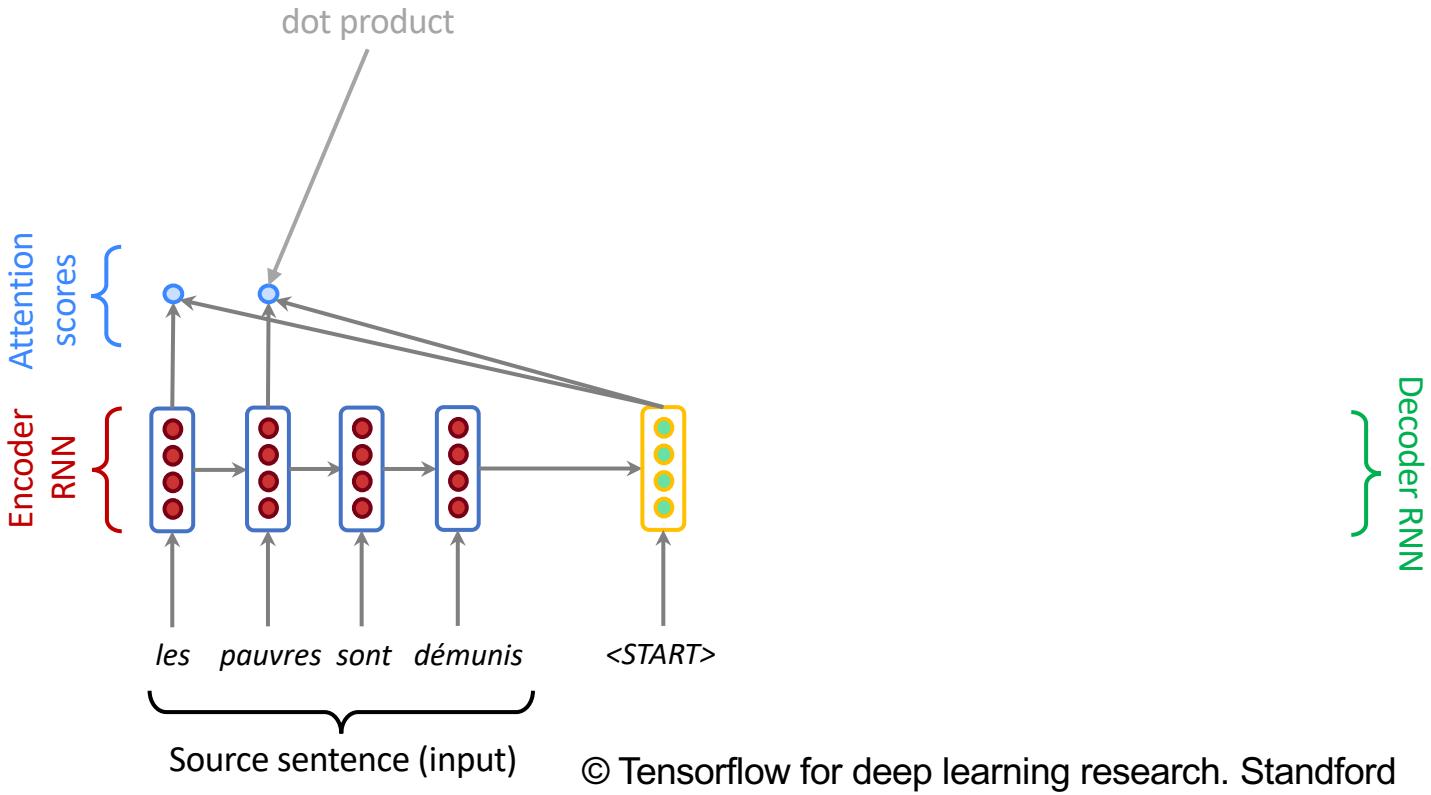
- **Attention** giải quyết vấn đề nút thắt cổ chai của seq2seq
- **Ý tưởng:** ở mỗi bước giải mã, sử dụng kết nối trực tiếp tới phần mạng mã hóa để tính toán và từ đó chỉ **tập trung (chú ý)** vào một phần cụ thể câu nguồn, bỏ qua những phần không liên quan.
- One of the most influential ideas in deep learning for NLP
  - Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

Sequence-to-sequence with attention



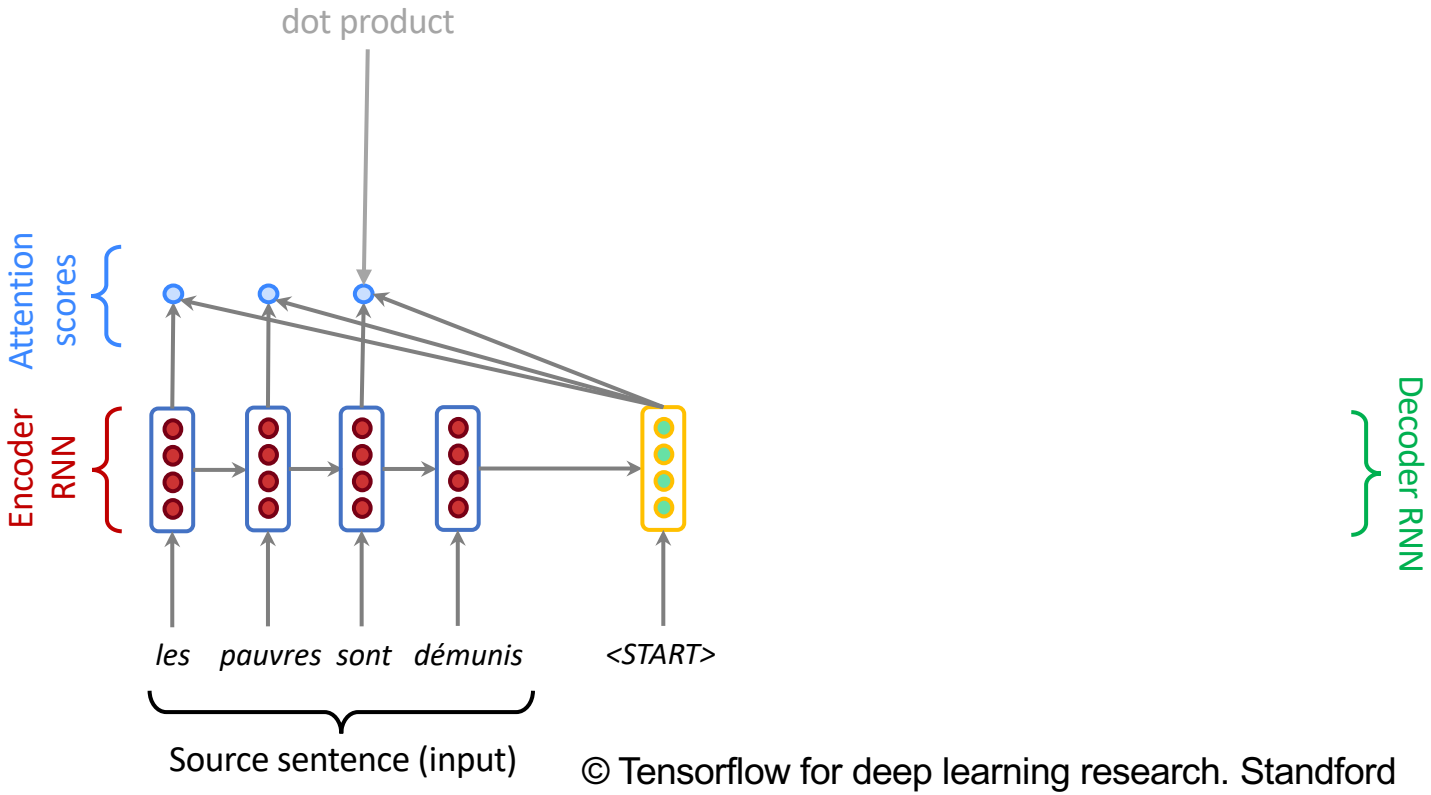
© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



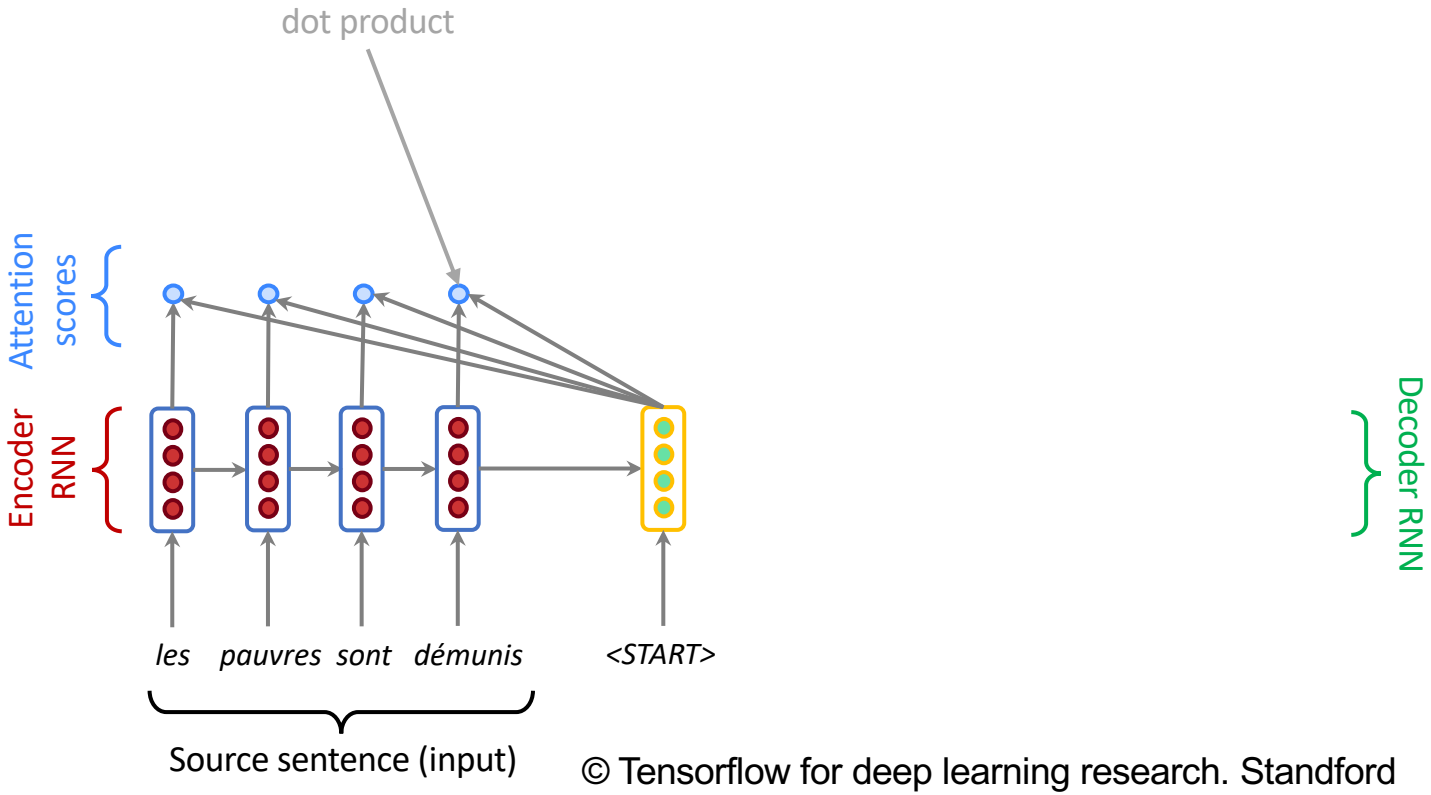
© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



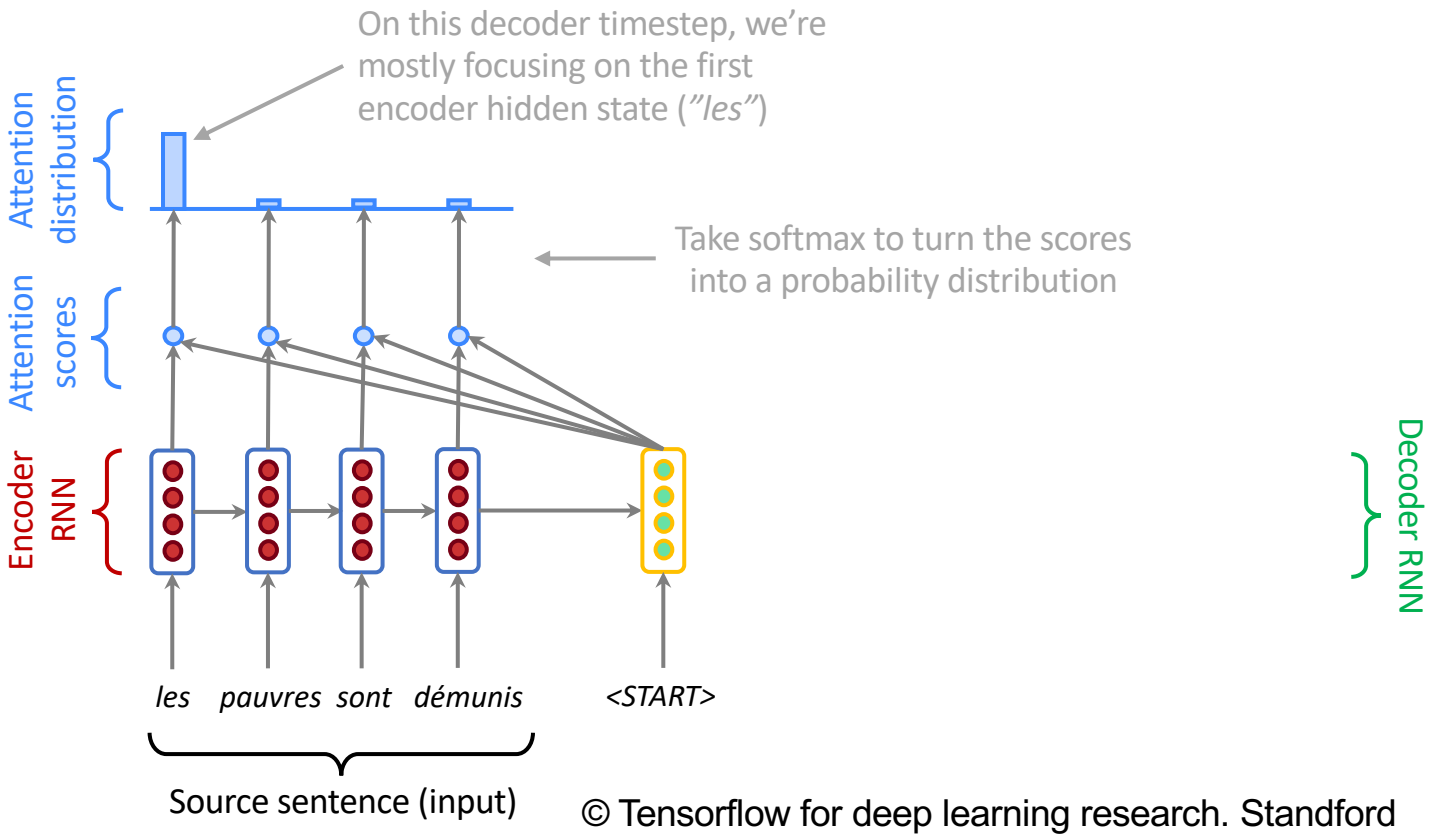
© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



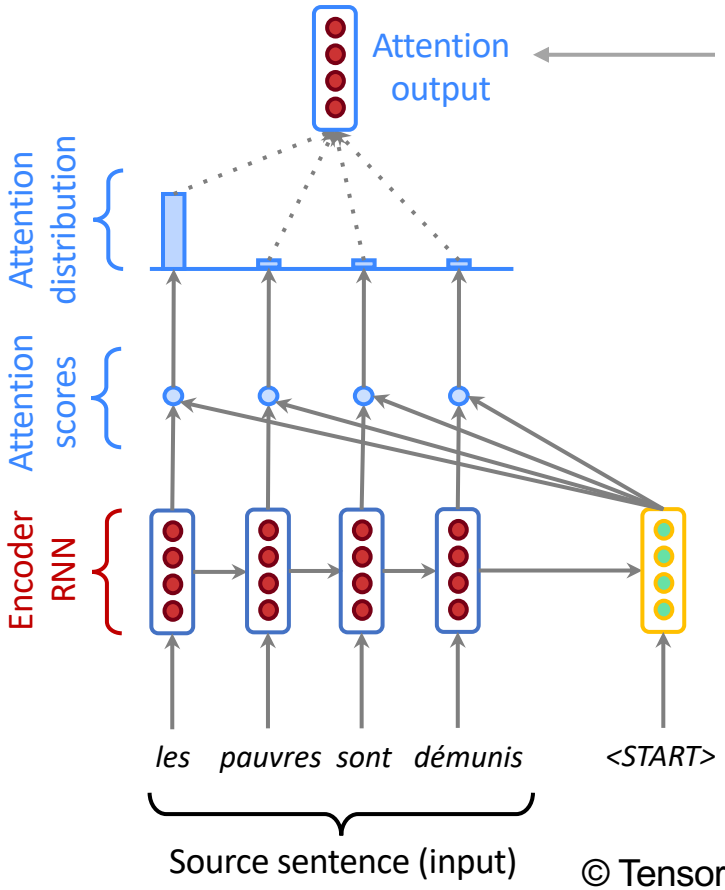
© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



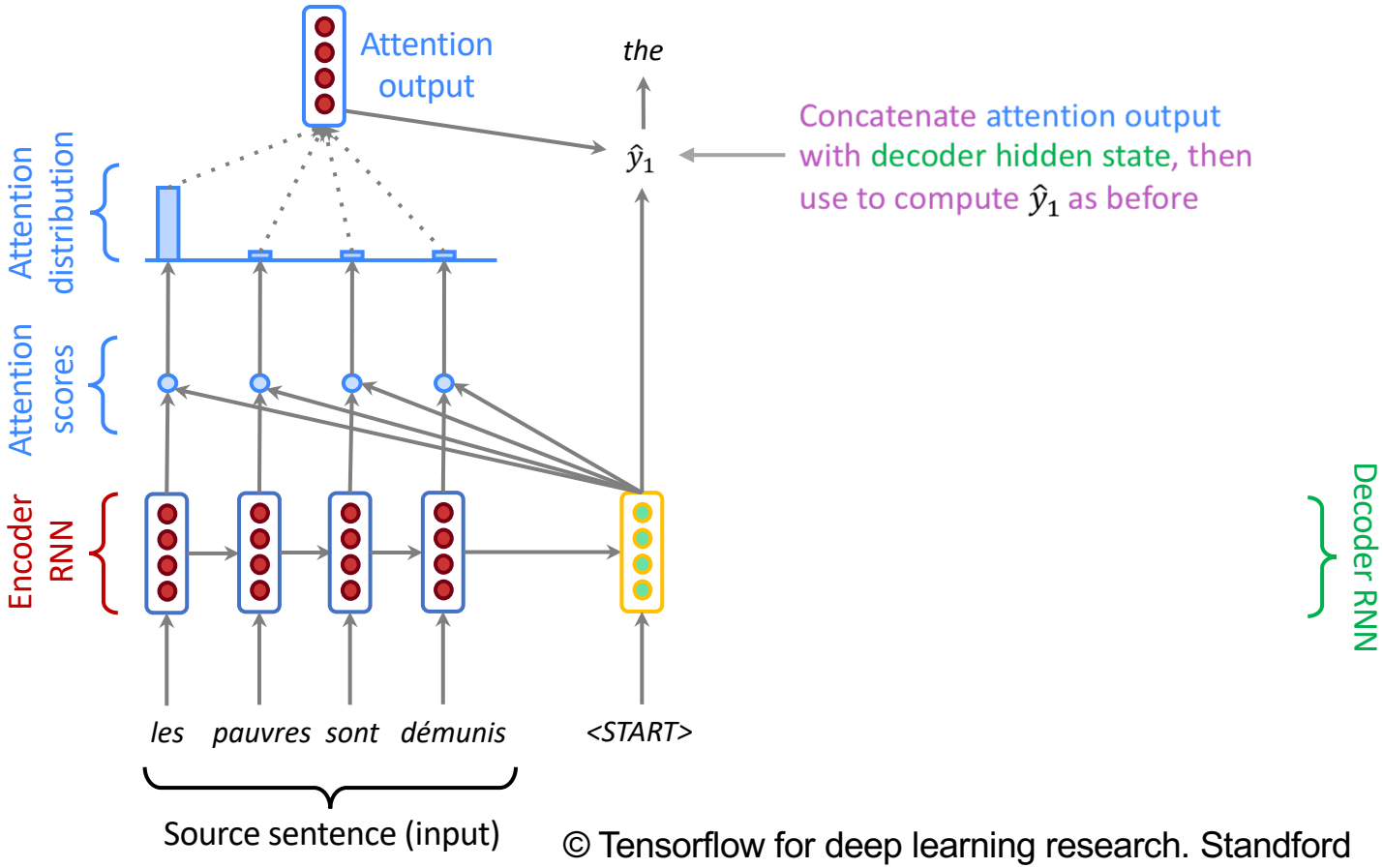
Use the attention distribution to take a **weighted sum** of the encoder hidden states.

The attention output mostly contains information the hidden states that received high attention.

Decoder RNN

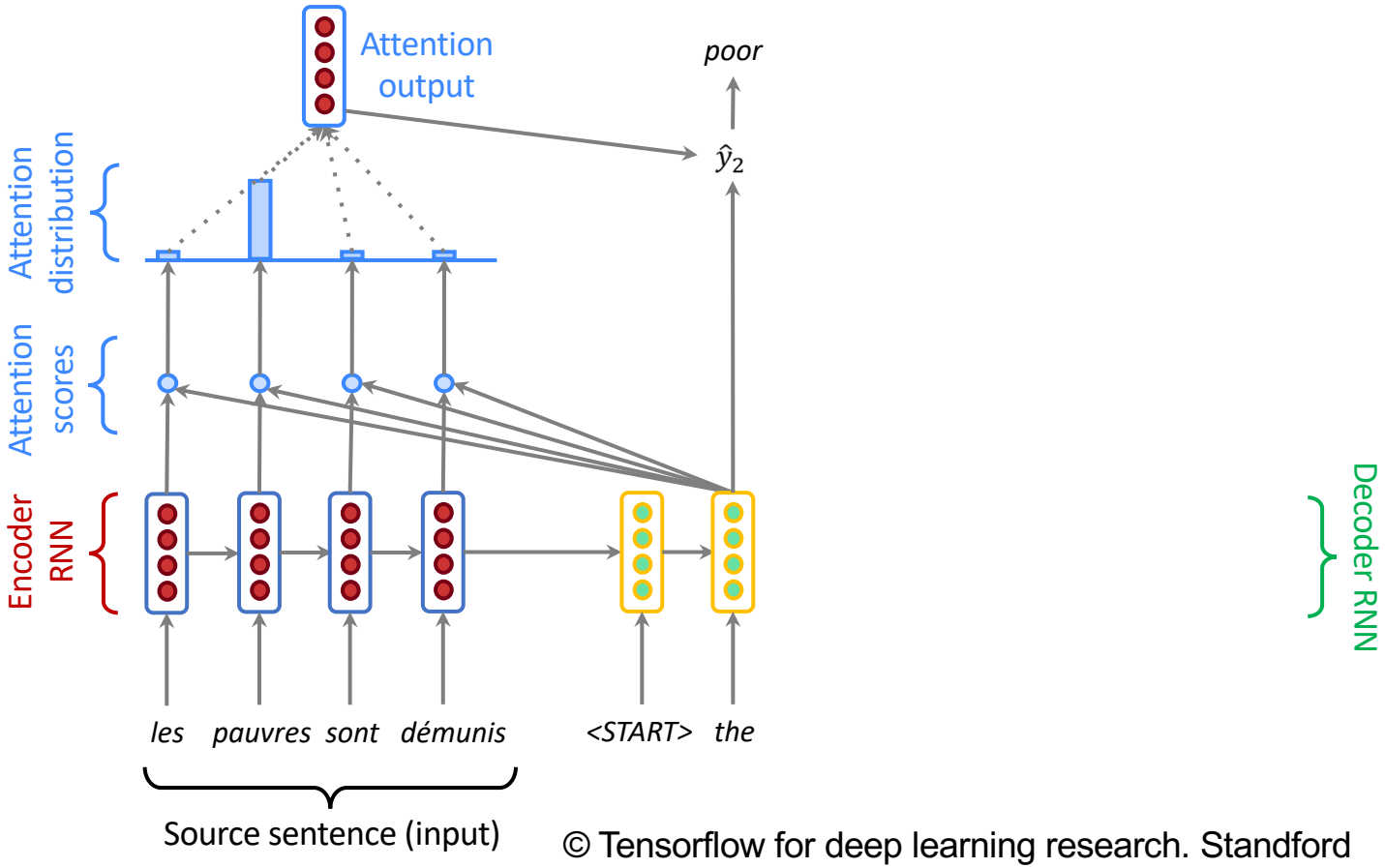
© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



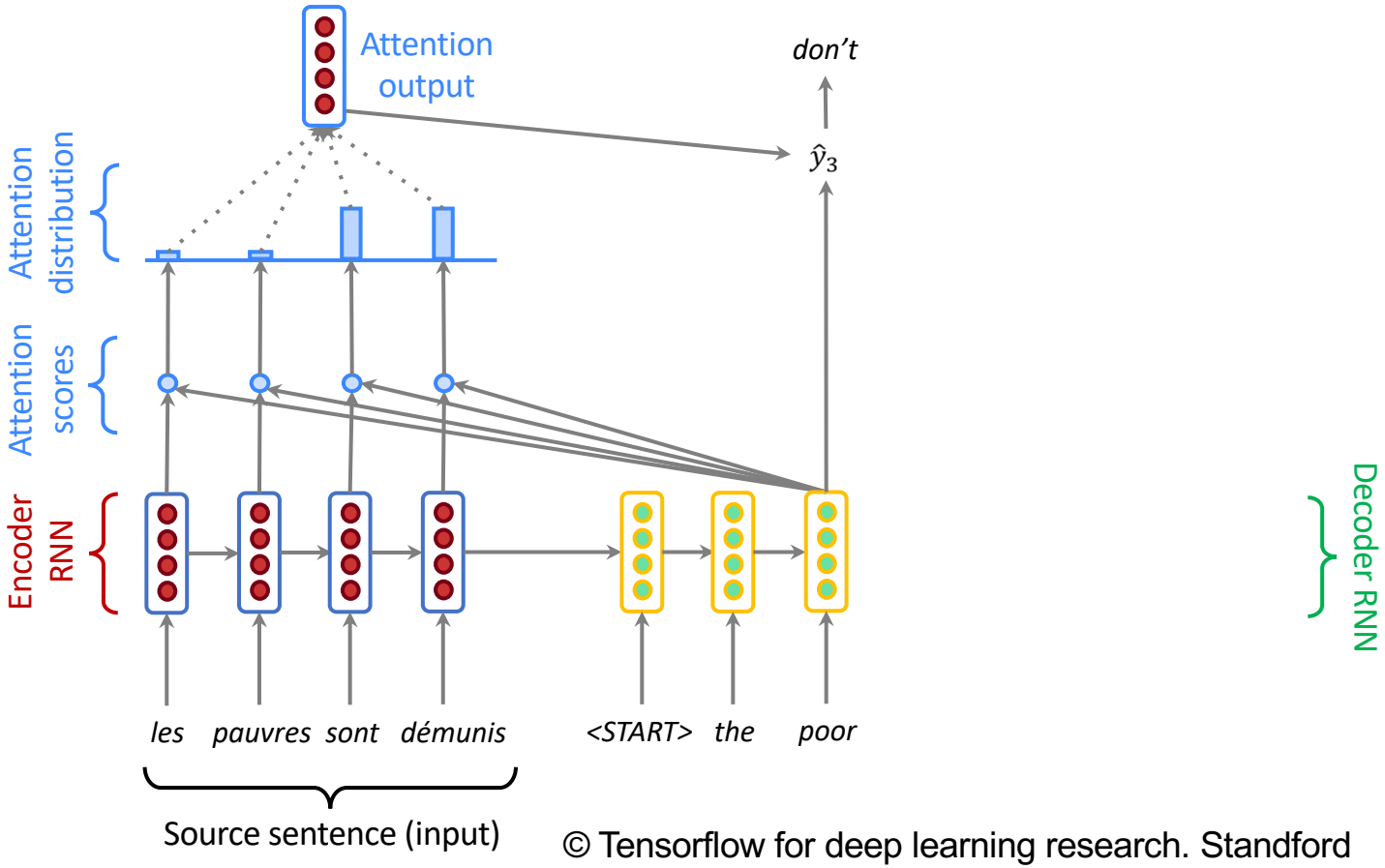
© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



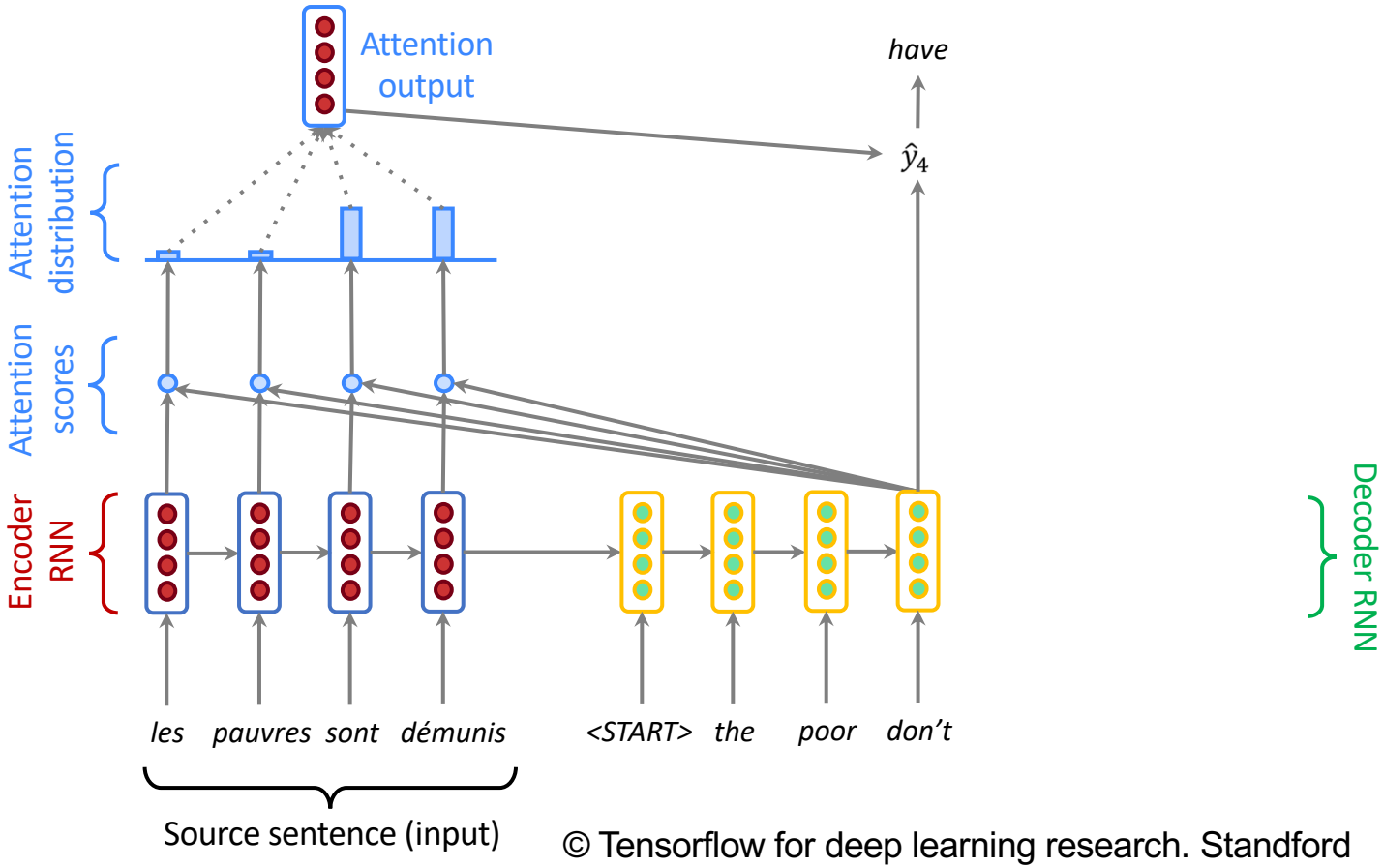
© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



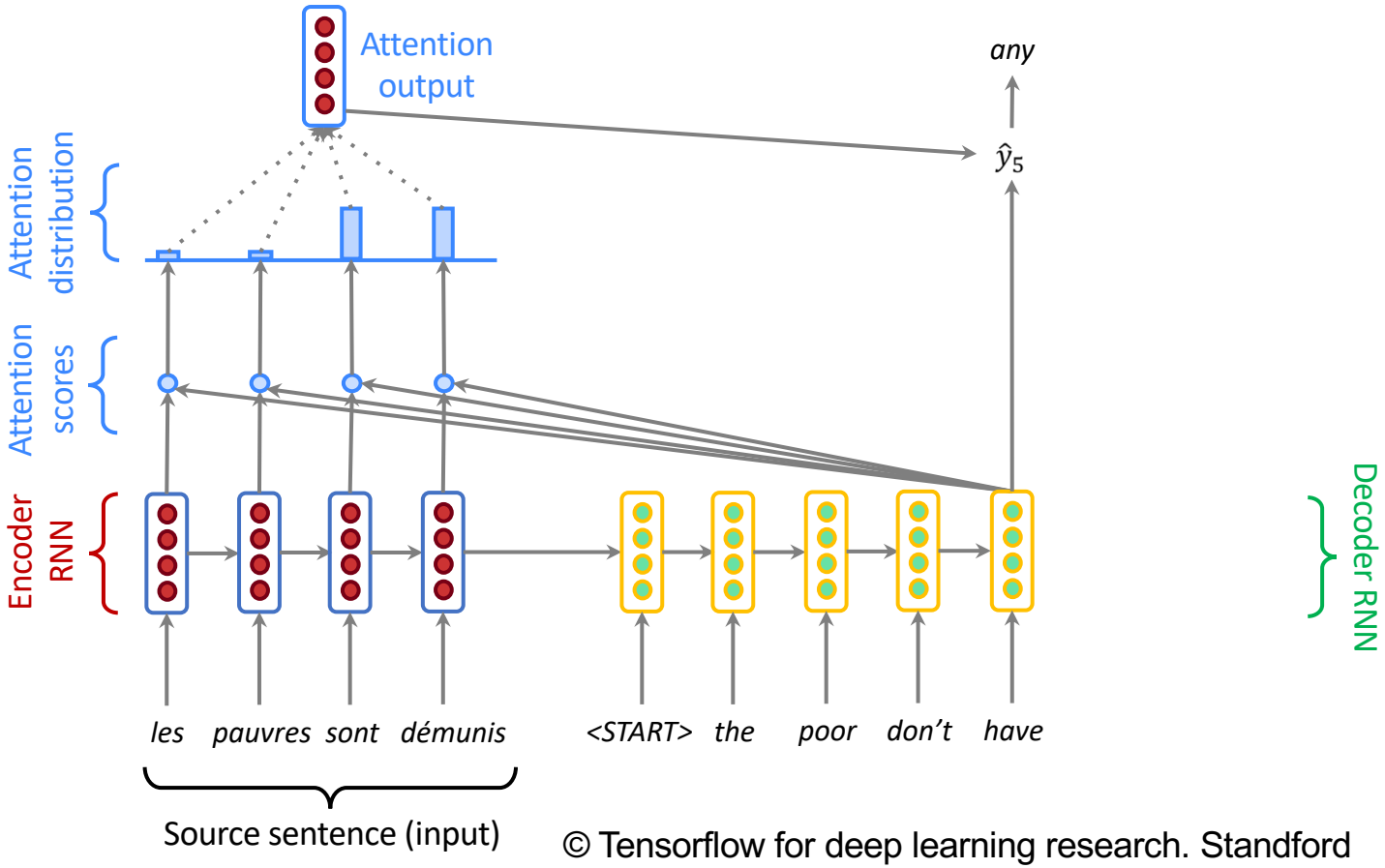
© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



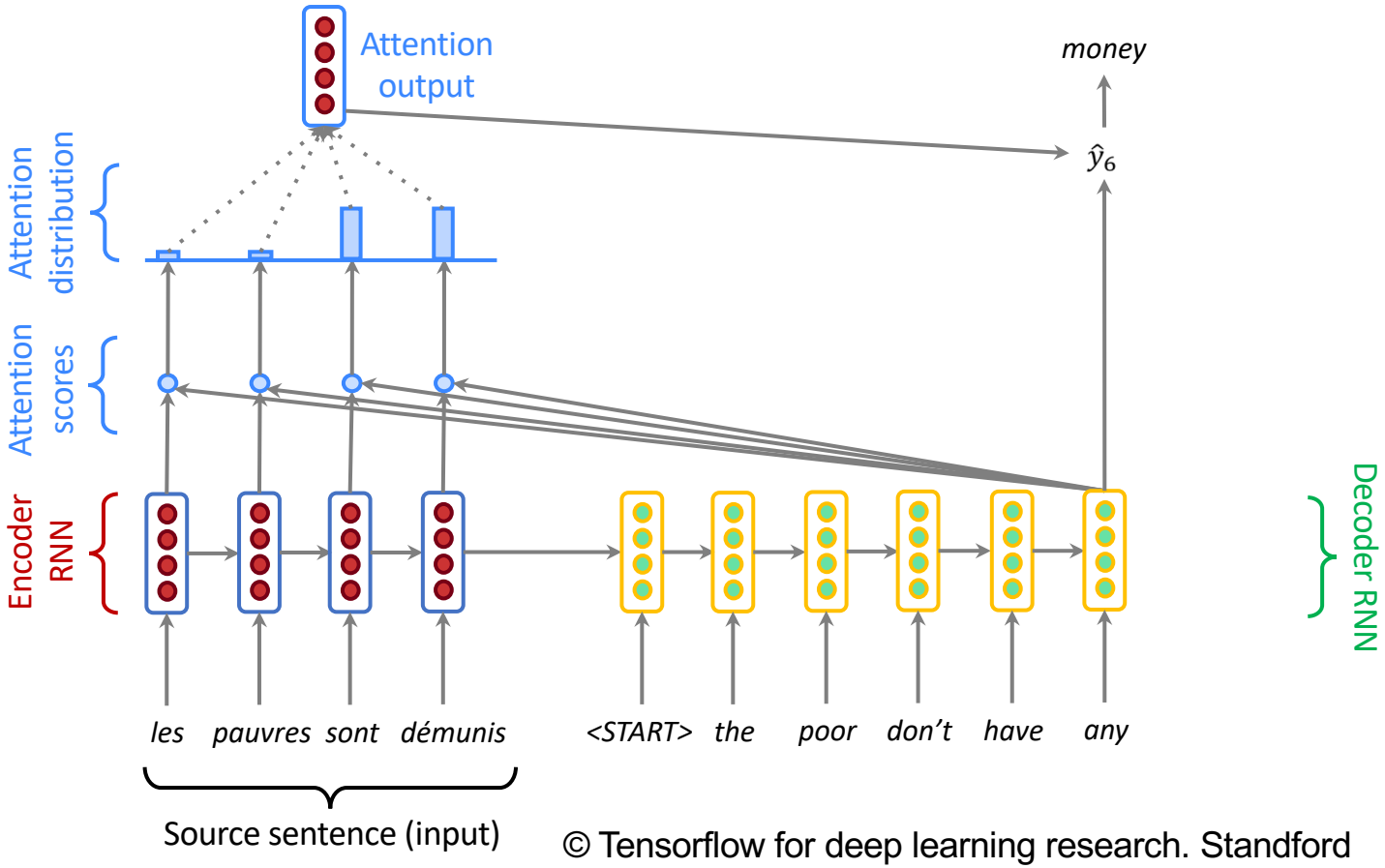
© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



© Tensorflow for deep learning research. Standford

Sequence-to-sequence with attention



© Tensorflow for deep learning research. Standford

Công thức chi tiết

- Mã hoá trạng thái ẩn $h_1, \ldots, h_N \in \mathbb{R}^h$
- Tại bước t, ta có trạng thái ẩn để giải mã $s_t \in \mathbb{R}^h$
- Điểm attention score e^t cho bước này:

$$\boldsymbol{e}^t = [\boldsymbol{s}_t^T \boldsymbol{h}_1, \ldots, \boldsymbol{s}_t^T \boldsymbol{h}_N] \in \mathbb{R}^N$$

- Tính softmax để có phân phối của sự chú ý α^t cho bước này (tổng phân phối xác xuất bằng 1)

$$\alpha^t = \text{softmax}(\boldsymbol{e}^t) \in \mathbb{R}^N$$

- Sử dụng α^t để tính tổng chập có trọng số của trạng thái ẩn của tầng encoder, mục tiêu để tính đầu ra của attention

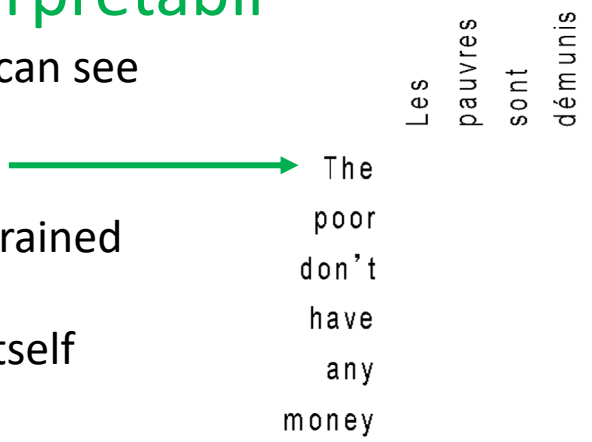
$$\boldsymbol{a}_t = \sum_{i=1}^{N} \alpha_i^t \boldsymbol{h}_i \in \mathbb{R}^h$$

- Gộp đầu ra của attention $\boldsymbol{a}_t$ với trạng thái ẩn của bộ giải mã decoder s_t , tiếp tục xử lý như mạng seq2seq thông thường

$$[\boldsymbol{a}_t; \boldsymbol{s}_t] \in \mathbb{R}^{2h}$$

Cơ chế chú ý có nhiều ưu điểm

- Attention significantly improves NMT performance
  - It's very useful to allow decoder to focus on certain parts of the source
- Attention solves the bottleneck problem
  - Attention allows decoder to look directly at source; bypass bottleneck
- Attention helps with vanishing gradient problem
  - Provides shortcut to faraway states
- Attention provides some interpretability
  - By inspecting attention distribution, we can see what the decoder was focusing on
  - We get alignment for free!
  - This is cool because we never explicitly trained an alignment system
  - The network just learned alignment by itself



© Tensorflow for deep learning research. Standford

Ứng dụng của mô hình seq2seq

- Summarization (long text → short text)
- Dialogue (previous utterances → next utterance)
- Parsing (input text → parse tree)
- DNA sequencing
- Voice recognition
- Text to speech

Tài liệu tham khảo

1. Khóa cs244n của Stanford:

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/

2. Khóa cs231n của Stanford:

http://cs231n.stanford.edu/slides/2020/lecture_10.pdf